

ESEIAAT  
MEMORIA PROJECTE FI DE GRAU



---

ESTUDI I DISSENY DE  
FILTRES DIGITALS

---

GRAU: Enginyeria de sistemes audiovisuals  
DATA D'ENTREGA: 10 de Juny del 2019  
NOM I COGNOMS D'ESTUDIANT: Mireia Gonzalez Ferrer  
NOM I COGNOMS DIRECTOR: Sisco Vallverdú



# INDEX

<b>TAULA DE FIGURES .....</b>	<b>4</b>
<b>1. INTRODUCCIÓ .....</b>	<b>4</b>
1.1. OBJECTE .....	7
1.2. ABAST .....	8
1.3. REQUERIMENTS .....	8
1.4. JUSTIFICACIÓ I UTILITAT .....	8
<b>2. DESENVOLUPAMENT .....</b>	<b>8</b>
2.1. ANTECEDENTS .....	8
2.1.1. Funcions de Matlab .....	8
2.1.2. Entorn gràfic Matlab .....	9
2.1.3. Filtres digitals .....	13
2.1.4. FILTER DESIGNER .....	17
2.2. PLANTEJAMENT .....	18
2.2.1. Suma de senyals.....	18
2.2.2. Filtres digitals .....	19
2.3. DESENVOLUPAMENT DE LA SOLUCIÓ.....	22
2.3.1. Suma de senyals.....	22
2.3.2. Filtres digitals .....	26
<b>3. RESULTATS .....</b>	<b>36</b>
3.1. PROVES .....	36
3.1.1. SUMA DE SENYALS .....	37
3.1.2. FILTRES .....	40
3.2. CONCLUSIONS I RECOMANACIONS .....	46
<b>4. BIBLIOGRAFIA I WEBGRAFIA.....</b>	<b>47</b>

## TAULA DE FIGURES

Figura 1: Representació de la pantalla GUIDE .....	10
Figura 2: Quadre de comandes Matlab, GUDE .....	10
Figura 3: Creació de GUI .....	11
Figura 4: Elements GUIDE .....	12
Figura 5: Toolbar editor .....	12
Figura 6: Callback .....	13
Figura 7: Tag .....	13
Figura 8: Filtre Digital .....	13
Figura 9: Tipus de filtres .....	15
Figura 10: Estructura directa filtre FIR .....	15
Figura 11: Estructura directa filtre IIR.....	16
Figura 12: FILTER DESIGNER .....	17
Figura 13: Plantejament de disseny de l'aplicació de suma de senyals..	18
Figura 14: Diagrama del funcionament de la aplicació de suma de senyals .....	19
Figura 15: Plantejament Menú Filtres digitals .....	20
Figura 16: Plantejament de entrada .....	20
Figura 17: Plantejament de disseny dels filtres .....	21
Figura 18: Funcionament filtrat de senyals .....	21
Figura 19: Disseny de suma de senyals .....	22
Figura 20: Inicialitzar figura .....	23
Figura 21: Funció set i lectura de variables.....	24
Figura 22: Funció MG_audio_senyal .....	24

Figura 23: Gràfica d'un senyal.....	25
Figura 24: Funció suma.....	25
Figura 25: Disseny del Menú.....	26
Figura 26: Callback per canviar de pantalla .....	28
Figura 27: Disseny d'Entrada .....	29
Figura 28: Funcionament tpg .....	31
Figura 29: Escollir archiu guardat al ordinador.....	31
Figura 30: Llegir archiu del ordinador.....	31
Figura 31: Disseny de Filtres.....	32
Figura 32: Disseny de toolbar de filtres .....	33
Figura 33: Codi de variacions de variables segons el filtre .....	35
Figura 34: Funció MG_trf .....	35
Figura 35: Gravar àudio 1 .....	37
Figura 36: Escoltar i gràfica d'àudio 1 i gravar àudio 2 .....	38
Figura 37: Escoltar i gràfica d'àudio 1 i àudio 2, botó suma .....	38
Figura 38: Escoltar i gràfica de la suma dels senyals .....	39
Figura 39: Repetir gravació d'àudio.....	39
Figura 40: Menu aplicació disseny de filtres .....	40
Figura 41: Pantalla d'entrada .....	40
Figura 42: Gravar àudio d'entrada.....	41
Figura 43: Penjar fitxer d'entrada .....	41
Figura 44: Elecció de punts per fer transformada de Fourier .....	42
Figura 45: Transformada de Fourier de la senyal d'entrada .....	42
Figura 46: Menú amb la senyal d'entrada .....	43

Figura 47: Disseny del filtre .....	43
Figura 48: Canvi de paràmetres en disseny de filtres .....	44
Figura 49: Gràfica en amplitud d'un filtre passa-baix IIR òptim.....	44
Figura 50: Menú amb la sortida.....	45

## 1. INTRODUCCIÓ

El present treball tracta de l'estudi i el disseny de filtres digitals, estudi en el qual es planteja la teoria dels diferents filtres que existeixen. En el cas d'aquest

projecte, els esmentats, són el FIR òptim i el IIR òptim, ja siguin passa-alt, passa-baix, passa-banda o elimina banda. Per altra banda el disseny, el que permetrà és crear un entorn gràfic per a poder implementar-los. Aquest projecte és molt important i rellevant, ja que en aquest moment ja existeix una aplicació que té la mateixa funció que la implementada, però la creada afegeix alguna funcionalitat, com poder filtrar senyals.

Matlab és el software amb què es treballa, amb la utilitat GUIDE en concret, el qual permet crear aplicacions de zero, amb les utilitats que es necessitin. Des d'una mirada subjectiva, es destaca que és una bona eina de programació. Per a poder utilitzar-la, s'ha hagut de fer una investigació del seu funcionament, en aquest cas amb bastant èxit, ja que es troba molt a l'abast de tothom i no té un ús gaire difícil.

A més, l'aplicació creada és molt interessant, ja que, a part del disseny de filtres, permet poder provar in situ el funcionament del filtratge, el qual permet realitzar més funcions bastant útils, com poder introduir un senyal d'entrada, filtrar-lo, i comparar-lo amb un senyal de sortida, amb la finalitat de poder visualitzar les diferències fàcilment, i guardar-lo.

El projecte en si pot ser una eina molt útil, que pot servir tant per l'aprenentatge d'alumnes a l'hora d'entendre la teoria dels filtres, com per a professionals amb objectius clars, per tal de modificar senyals d'àudio perquè s'escoltin molt millor.

Podria estar a l'abast de tothom, ja que els passos a seguir es troben ben definits i orientats a obtenir un bon funcionament.

## 1.1. OBJECTE

L'objecte d'estudi es basa en l'aprenentatge del funcionament d'un entorn gràfic de Matlab, el qual ens permet realitzar aplicacions diverses. Per aconseguir

aquest propòsit, s'ha realitzat una aplicació anterior al treball, la qual permetia la suma de dos senyals diferents i la representació del resultat.

Una vegada assolits els coneixements, el que es pretén és poder realitzar un altre tipus d'aplicació, la qual permet crear filtres digitals, introduint diferents paràmetres, així com filtrar un senyal d'entrada per obtenir un de sortida.

## 1.2. ABAST

L'objectiu d'aquest projecte es tracta de poder crear una aplicació fent servir GUIDE del Matlab, que faci possible el filtratge d'àudios d'entrada, ja siguin gravats en el mateix moment com penjats des de l'ordinador, amb el filtre que més convingui, proporcionant una sortida adequada.

## 1.3. REQUERIMENTS

Per a la realització del projecte és necessari conèixer el llenguatge per a programar en Matlab, adquirir els coneixements per a crear l'entorn gràfic de Matlab, saber el comportament dels diferents tipus de filtre, i per últim saber implementar-los per tal de poder utilitzar-los amb un senyal d'entrada.

## 1.4. JUSTIFICACIÓ I UTILITAT

Aquest entorn gràfic de disseny de filtres ja està dissenyat per Matlab, anomenat Filter Designer, però el seu funcionament tan sols té com a objectiu obtenir els coeficients del filtre, per això el que es pretén és afegir, a part del disseny, el filtratge de senyals, el qual pot ser utilitzar per a qualsevol usuari amb els coneixements o no de filtrat de senyals.

# 2. DESENVOLUPAMENT

## 2.1. ANTECEDENTS

### 2.1.1. Funcions de Matlab

Matlab conté diverses funcions creades que es poden utilitzar amb el simple fet de cridar-les, en aquest projecte es preveu que s'utilitzaran, entre d'altres bàsiques, les següents permeten:



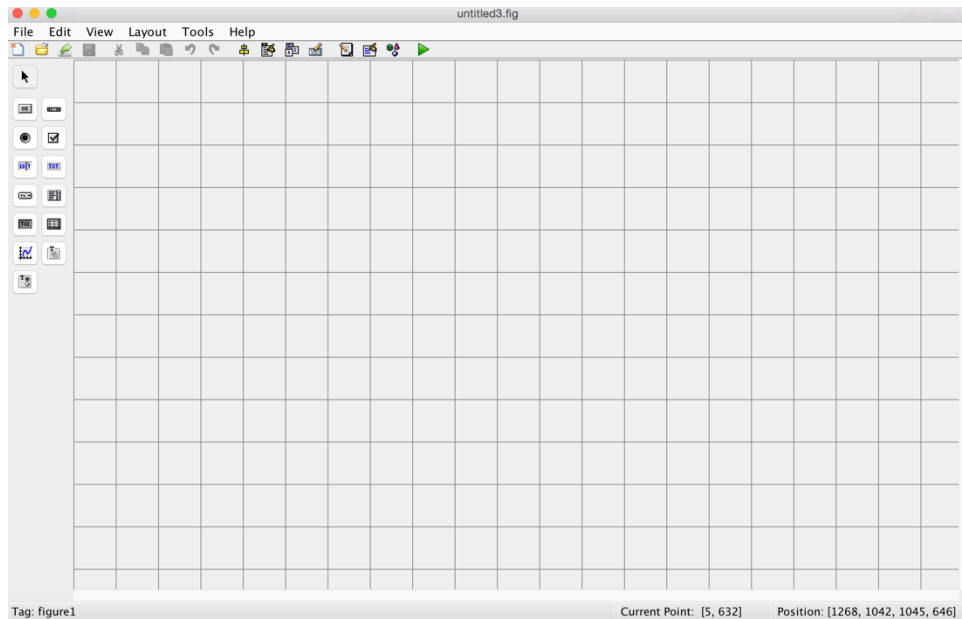
- *'audiorecoder'*: Gravar l'àudio amb micròfon, com a paràmetres d'entrada es tenen el temps i la freqüència de mostratge, i com a sortida el senyal gravat.
- *'plot'*: Representar un senyal en una gràfica.
- *'uigetfile'*: Obtenir el destí i el nom d'un fitxer triat a l'explorador de l'ordinador.
- *'get'*: Obtenir un valor d'un element del GUIDE.
- *'audioread'*: Llegir un àudio guardat a l'ordinador, com a paràmetres d'entrada s'introdueixen el nom i el destí i com a sortida el senyal.
- *'ginput'*: Marcar amb el cursor un punt d'una gràfica, el qual permet guardar la seva posició.
- *'fft'*: Realitzar la transformada de Fourier d'un senyal.
- *'set'*: Modificar qualsevol element de GUIDE.
- *'conv'*: Convolució de dos senyals.
- *'sound'*: Reproduir un senyal d'àudio.

### 2.1.2. Entorn gràfic Matlab

Primerament per poder dur a terme el projecte s'han assolit els coneixements del funcionament de la creació d'aplicacions amb interfases gràfiques de usuari de Matlab, anomenades GUI.

Les GUI permeten un control senzill de les aplicacions de software, per tant per poder-les utilitzar no és necessari aprendre un llenguatge o escriure comandes per a poder executar una aplicació.

Les apps de Matlab són programes autònoms amb un frontal gràfic d'usuari GUI que automatitzen qualsevol feina o càlcul, amb controls com menús, botons, entre d'altres.

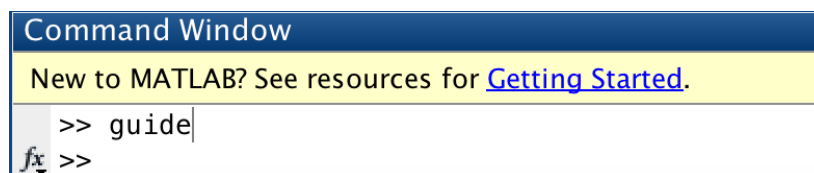


*Figura 1: Representació de la pantalla GUIDE*

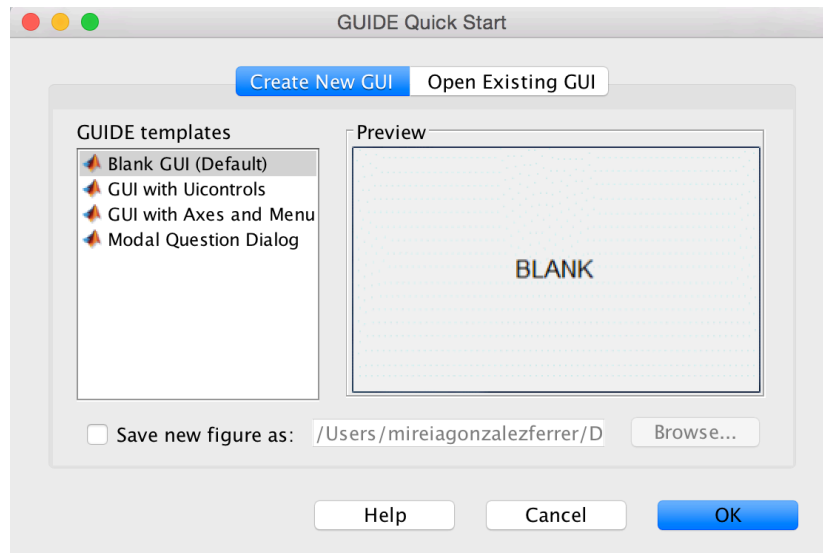
GUIDE és l'entorn de desenvolupament de GUI, tal com es veu a la figura 1. Aquest proporciona eines per a dissenyar interfases d'usuari per aplicacions personalitzades, com és el cas d'aquest projecte.

A l'utilitzar aquest editor, es genera automàticament el codi de Matlab per a construir la interfase, el qual es pot modificar depenen del comportament que es vulgui que tingui cada component.

Per la creació de la GUI, cal obrir GUIDE amb la comanda "guide" al quadre de comandes de Matlab, tal com es veu a la figura 2, i triar l'opció "Blank GUI default", com es mostra a la figura 3.



*Figura 2: Quadre de comandes Matlab, GUIDE*



*Figura 3: Creació de GUI*

Com es pot veure a la part esquerra de la il·lustració 1, tenim una barra d'eines la qual permet poder escollir quin tipus d'element es vol utilitzar en la aplicació, a la figura 4 es poden veure cadascun d'ells:

- StaticText: Text estàtic, el qual serveix com a definició de quin paràmetre s'està introduint.
- EditText: Text editable, es qual podem canviar segons ens convingui.
- Pushbutton: Botó, que realitza alguna funció pressionant-lo.
- Pop-up Menú: Menú desplegable, en el qual podem triar qualsevol de les opcions.
- Axes: Quadre de gràfica, on es representarà el que es necessiti.

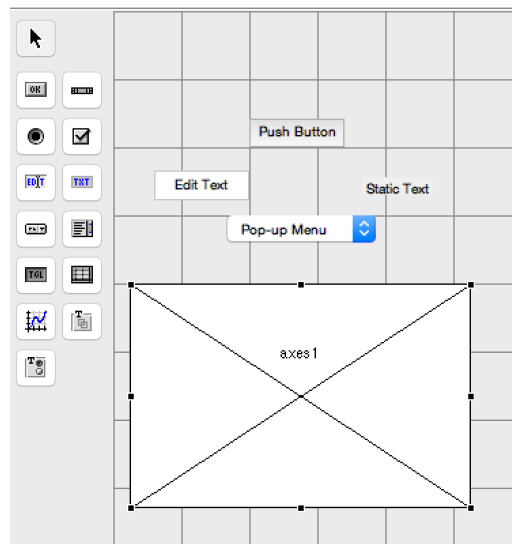


Figura 4: Elements GUIDE

A part d'aquests elements, també tenim l'opció de modificar la barra d'eines de la part superior de les figures, anomenat Toolbar Editor, tal com es mostra a la figura 5, el que permet és crear les icones del que voldrem incloure.

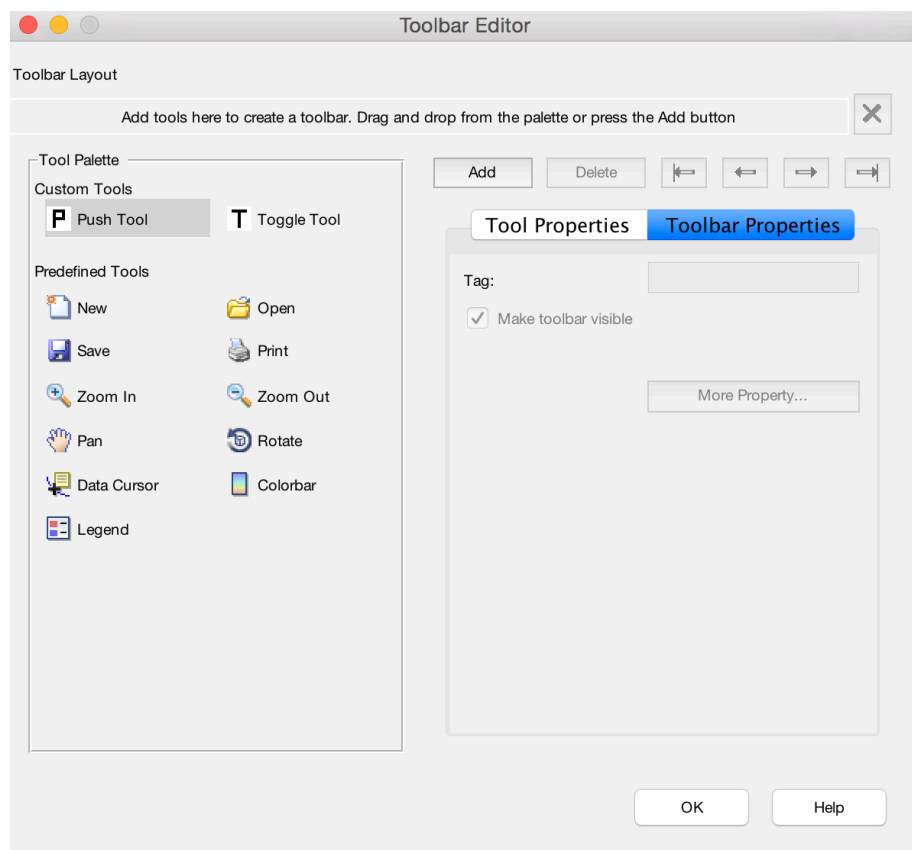


Figura 5: Toolbar editor

A mesura que es va dissenyant l'aplicació, es van generant de manera automàtica diferents funcions per a cada element. De cadascun d'ells, es poden modificar diferents paràmetres, un d'ells és el callback, tal com es pot veure a la figura 6. Aquest ens permet introduir un codi pel correcte funcionament de l'aplicació, tanmateix també és possible assignar un TAG per a cadascun, com apareix a la figura 7. D'aquesta manera, és possible cridar els elements als nostres 'scripts' generats, per tal d'aconseguir que l'aplicació finalment realitzi el comportament que es vol.

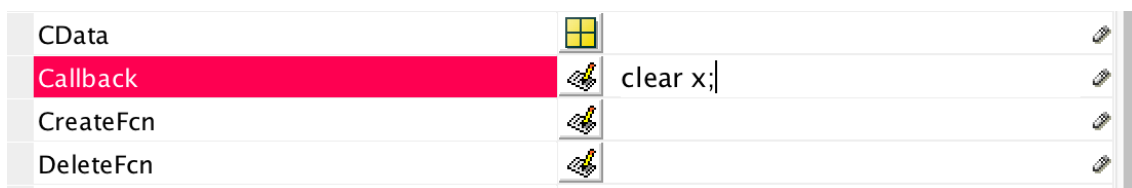


Figura 6: Callback

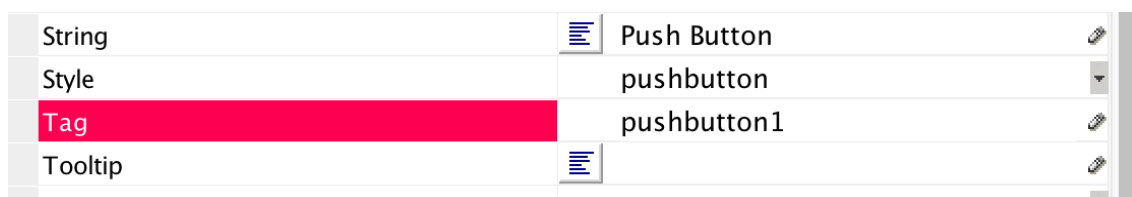


Figura 7: Tag

### 2.1.3. Filtres digitals

Un filtre digital és un sistema que realitza un processament matemàtic sobre un senyal, obtenint una sortida.

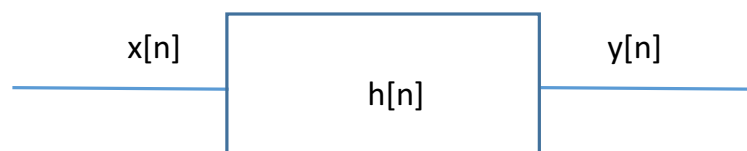


Figura 8: Filtre Digital

Tal com es mostra a la figura 8, els filtres digitals tenen com a entrada un senyal digital i com sortida un altre senyal digital que pot haver canviat en amplitud, freqüència o fase, depenent de les característiques del filtre digital.

Considerant un sistema lineal invariant temporal amb relació entrada-sortida definida per la següent equació, en la qual la sortida depèn de les mostres actuals i passades de l'entrada i les passades de la sortida:

$$y[n] = \sum_{i=0}^M b_i x[n-i] - \sum_{i=1}^P a_i y[n-i]$$

Els coeficients són a i b, són els que defineixen el filtre, per tant el disseny d'aquest consisteix en calcular-los.

$h[n]$  és la resposta impulsional del filtre digital, la qual permet expressar la relació entrada-sortida a partir de l'equació de convolució:

$$y[n] = x[n] * h[n] = \sum_{m=0}^{\infty} h[m]x[n-m]$$

La funció de transferència és  $H(z)$ :

$$H(z) = \sum_{n=0}^{\infty} h[n]z^{-n} = \frac{B(z)}{A(z)} = \frac{\sum_{n=0}^M b_n z^{-n}}{1 - \sum_{n=1}^P a_n z^{-n}}$$

Existeixen diversos tipus de filtres digitals, en aquest projecte, els filtres utilitzats, es poden diferenciar segons la part de l'espectre que atenuen o deixen passar, que són filtres passa-baix, passa-alts, passabanda i elimina-banda, i també segons el tipus de resposta davant l'entrada, els FIR i els IIR.

- PASSA-BAIX ("Lowpass"): Es tracta d'un filtre que deixa passar les freqüències baixes i atenua les altes.
- PASSA-ALT ("Highpass"): És el filtre que atenua les freqüències baixes i deixa passar les altes.

- PASSA-BANDA (“Bandpass”): Aquest deixa passar les freqüències que es troben en un rang.
- ELIMINA-BANDA (“Bandstop”): S’encarrega d’atenuar les freqüències que es troben en un rang.

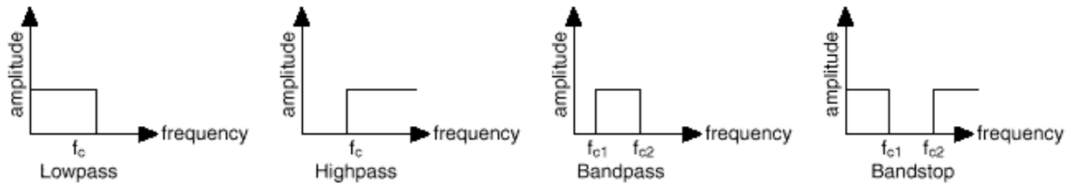


Figura 9: Tipus de filtres

- FIR: Es tracta d’un sistema lineal i invariant, caracteritzat per una resposta impulsional causal diferent de zero de longitud  $L$  mostres. L’ordre d’un filtre d’aquest tipus seria  $M=L-1$ .

L’estructura directa d’un filtre FIR seria la següent:

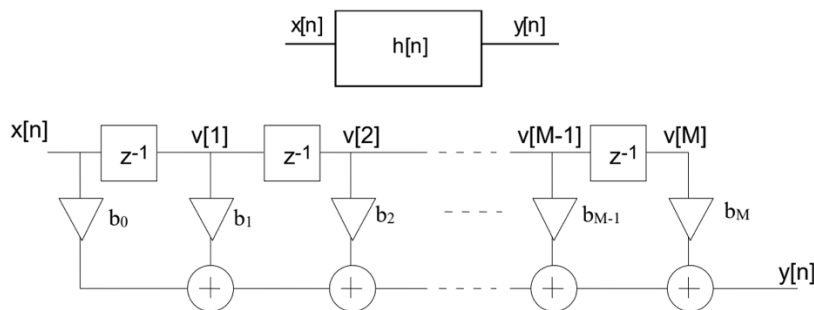


Figura 10: Estructura directa filtre FIR

Per cada mostra d’entrada es pot calcular la mostra de sortida realitzant el producte vectorial entre el vector de coeficients  $b$  i el vector de senyal  $x_n$ . Un cop obtingut el càlcul de la mostra de sortida, s’actualitza el vector del senyal amb la nova mostra d’entrada i es calcula la nova sortida. El procés s’itera mentre hi hagi senyal a l’entrada del filtre.

El vector de senyal està format per la mostra actual a l’entrada i les “ $M$ ” mostres passades.

En el cas d'aquest projecte serà un filtre FIR òptim, Algorisme de Parks-McClellan, el qual proporciona els coeficients d'un filtre FIR de mínim ordre, aproximant un comportament d'arissament constant a les bandes de pas i atenuades.

El filtre s'obté mitjançant l'algorisme de Remez, desenvolupat per Parks i McClellan. Aquest algorisme calcula el conjunt de  $L=M+1$  coeficient d'un filtre FIR amb la millor aproximació de la resposta freqüencial a l'especificada com ideal. Si l'aproximació no és prou bona, caldrà augmentar el valor de "M", en cas contrari cal disminuir-lo. El procediment és iteratiu fins a assolir l'ordre adequat.

- IIR: És un sistema lineal i invariant, caracteritzat per una resposta impulsional que es correspon a un sistema retroalimentat d'ordre "P". La longitud en resposta impulsional és infinita. L'ordre del filtre seria "Q", valor màxim entre l'ordre del numerador "M2 i el del denominador "P", de la funció de transferència  $H(z)$ .

L'estructura directa seria la següent:

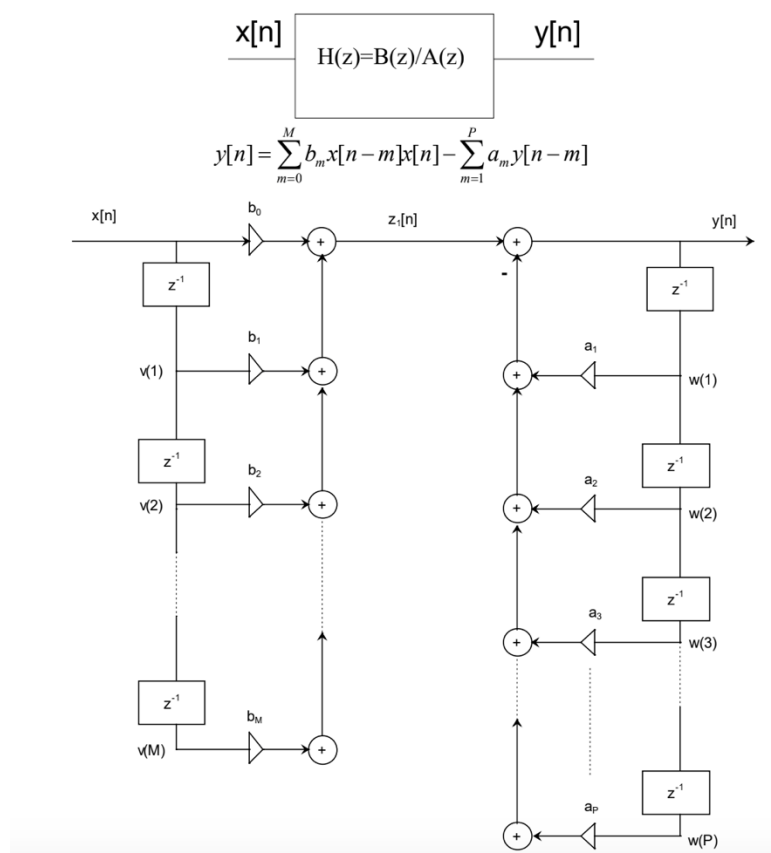


Figura 11: Estructura directa filtre IIR



L'esquema és la implementació directa de la relació entrada-sortida. La primera part correspon a una estructura FIR,  $h_1[n]$  i la segona a una estructura realimentada,  $h_2[n]$ . Aquesta estructura té  $M+P+2$  sumadors,  $M+P$  multiplicadors i  $M+P$  registres.

El filtre utilitzat al projecte és el IIR òptim, Algorisme de Cauer o el·líptic Matlab, que incorpora les funcions de càlcul d'aquest tipus de procediment. En el cas de fer servir l'aproximació de Cauer es fan servir les funcions ellipord i ellip.

#### 2.1.4. FILTER DESIGNER

Tal com s'ha esmentat anteriorment es tracta de realitzar una aplicació de filtres digitals, en aquest cas ja està creada dins de les aplicacions incloses a Matlab, anomenada FILTER DESIGNER. El seu funcionament és el mateix que l'aplicació del projecte.

En la següent figura es pot observar com és l'aplicació, i el seu funcionament. A la part de dalt a l'esquerra es troba un quadre, on es pot visualitzar les característiques dels filtres creats en tot moment. A la part de dalt a la dreta es troba la gràfica, la qual varia en funció del que es vol veure representat. En aquest apartat poden utilitzar l'opció: amplitud, fase, resposta impulsional, pols i zeros, entre d'altres.

Tot i això, a la part inferior trobem l'espai on s'introdueixen els paràmetres que es necessiten per la creació del filtre, donant l'opció de triar entre els diferents tipus, introduint les freqüències, d'amplitud i ordre segons convingui.

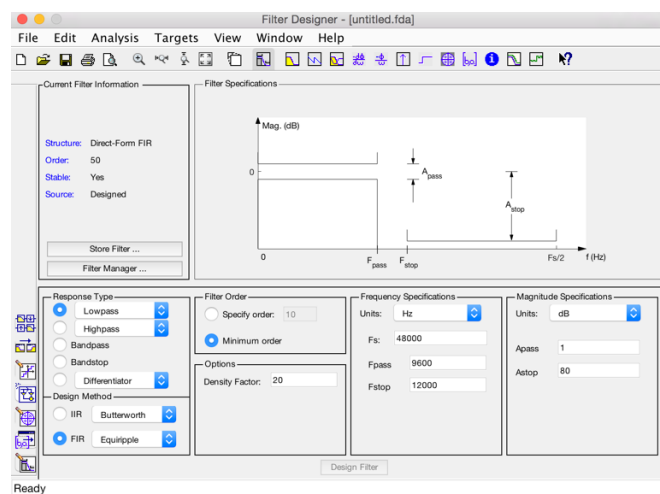


Figura 12: FILTER DESIGNER

## 2.2. PLANTEJAMENT

### 2.2.1. Suma de senyals

L'inici del desenvolupament del projecte és la creació d'una aplicació, per assolir els coneixements de com crear una aplicació amb el GUIDE de Matlab.

Aquesta aplicació consisteix en la creació d'un entorn gràfic amb el qual sigui possible la suma de dos senyals que es graven amb un micròfon.

També es podran escoltar tant les senyals individuals com la suma d'elles, i per últim, es podrà visualitzar la representació de cadascuna d'elles.

El primer pas seria fer el plantejament del disseny de l'aplicació, pensant en els elements necessaris pel correcte funcionament, el qual es pot veure a la següent figura:



Figura 13: Plantejament de disseny de l'aplicació de suma de senyals

Després es planteja el funcionament de l'aplicació:

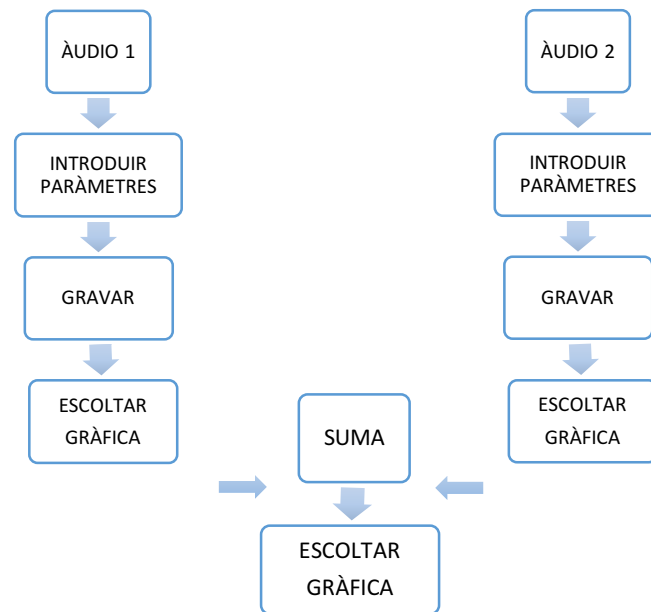


Figura 14: Diagrama del funcionament de la aplicació de suma de senyals

### 2.2.2. Filtres digitals

L'aplicació que es vol crear està dividida en dues funcions, una es tracta de la creació de filtres que és similar a Filter Designer de Matlab i l'altre estaria encaminada a introduir un àudio d'entrada i obtenir una sortida gràcies al filtre creat. Per tant es plantegen tres dissenys.

En el primer el que es pretén fer és un menú en el qual es podrà entrar a entrada, a filtre i finalment, un cop s'obtinguin les dades, es podrà veure tant l'entrada com la sortida representades, en amplitud i amb la transformada de Fourier i es podran escoltar.

També serà possible guardar el senyal de sortida.

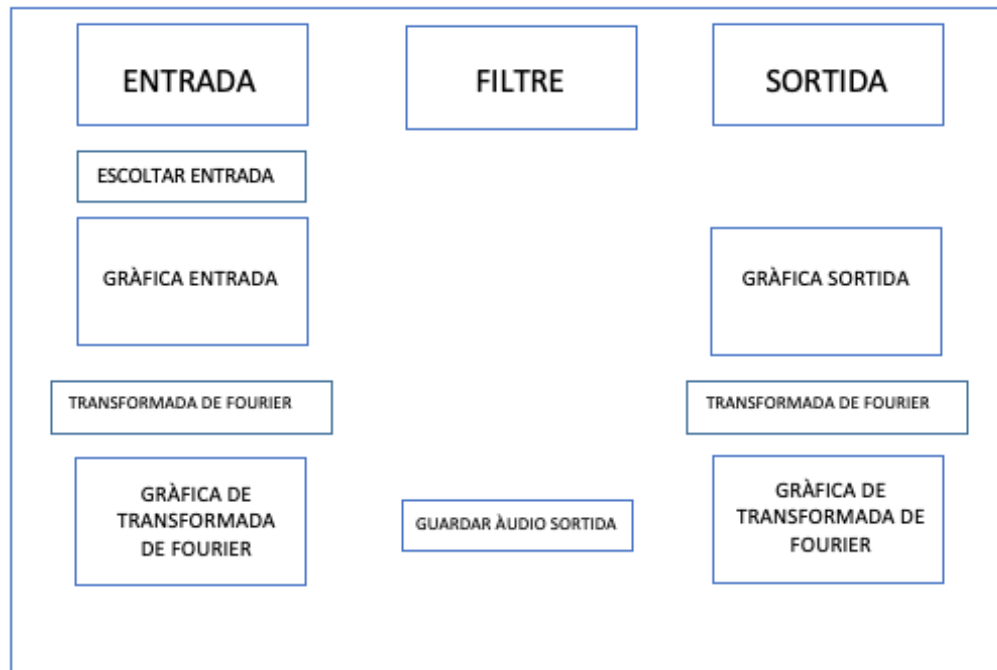


Figura 15: Plantejament Menú Filtres digitals

Seguidament, es fa una pantalla en la qual es pot entrar des del menú. Dins d'aquesta és possible gravar un àudio en el moment, especificant els paràmetres, o també és possible penjar un arxiu d'àudio que es trobi en el PC utilitzat. Una vegada ja es tenen les mostres d'entrada es representen en amplitud, en la transformada de Fourier i s'escolta.

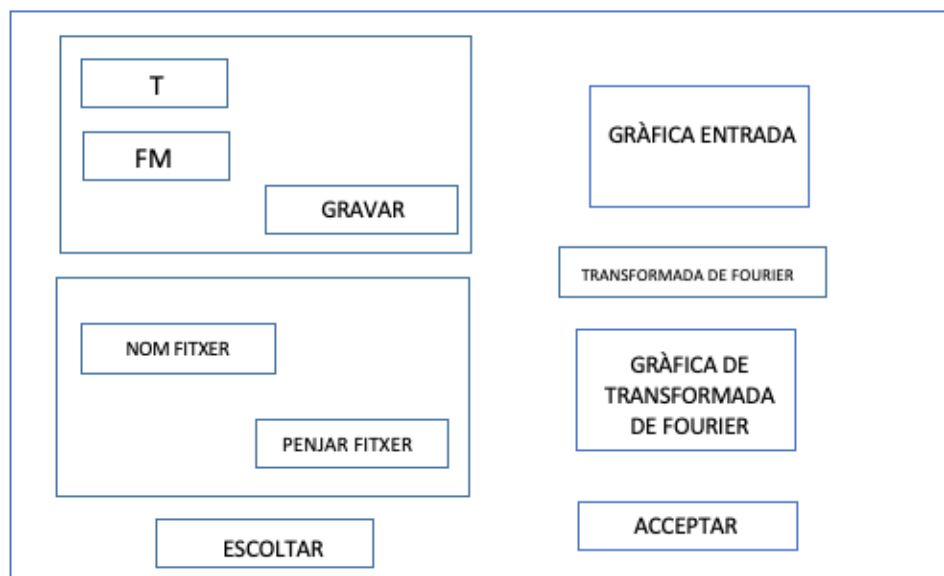


Figura 16: Plantejament de entrada

Per últim es planteja el disseny dels filtres en el qual hi haurà un quadre amb la informació

del filtre, dos elements on es podran triar els tipus de filtres, un quadre de paràmetres on s'introduiran els necessaris i una gràfica en la qual es podrà representar el filtre.

Figura 17: Plantejament de disseny dels filtres

El diagrama del funcionament de l'aplicació serà el següent:

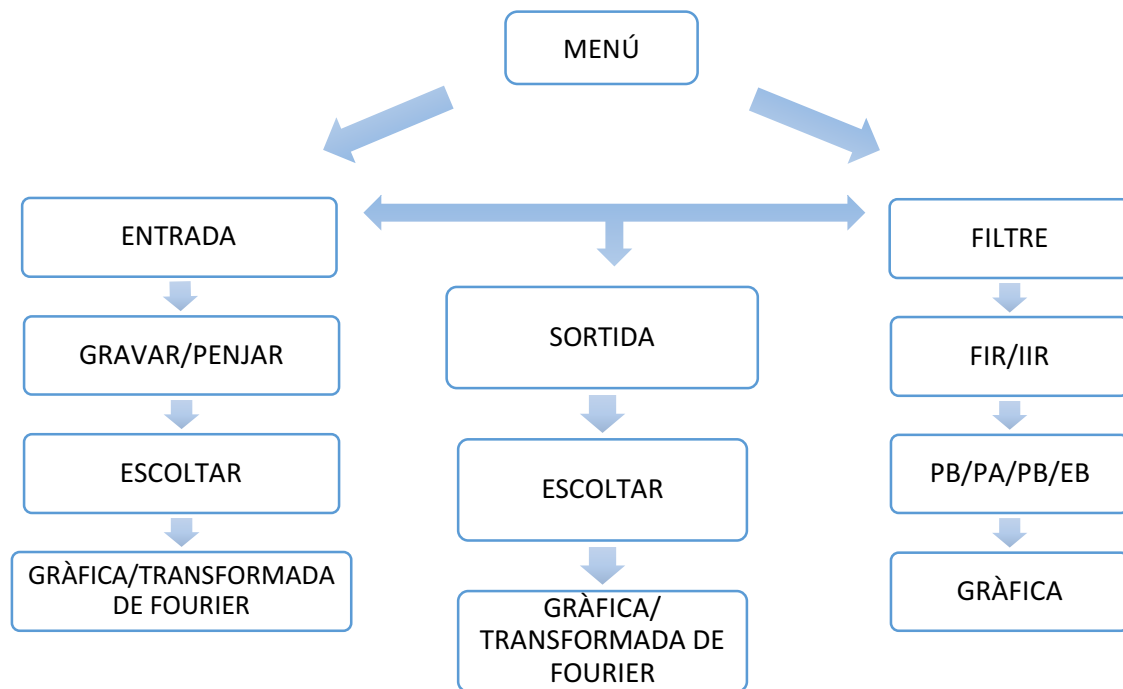


Figura 18: Funcionament filtrat de senyals

## 2.3. DESENVOLUPAMENT DE LA SOLUCIÓ

### 2.3.1. Suma de senyals

Per començar a dissenyar l'aplicació de suma de senyals, es van crear els directoris i fitxers que es necessitarien. És el cas d'una figura, anomenada 'MG\_sum\_gui.fig', que és la que es mostra a continuació, formada pels elements necessaris.

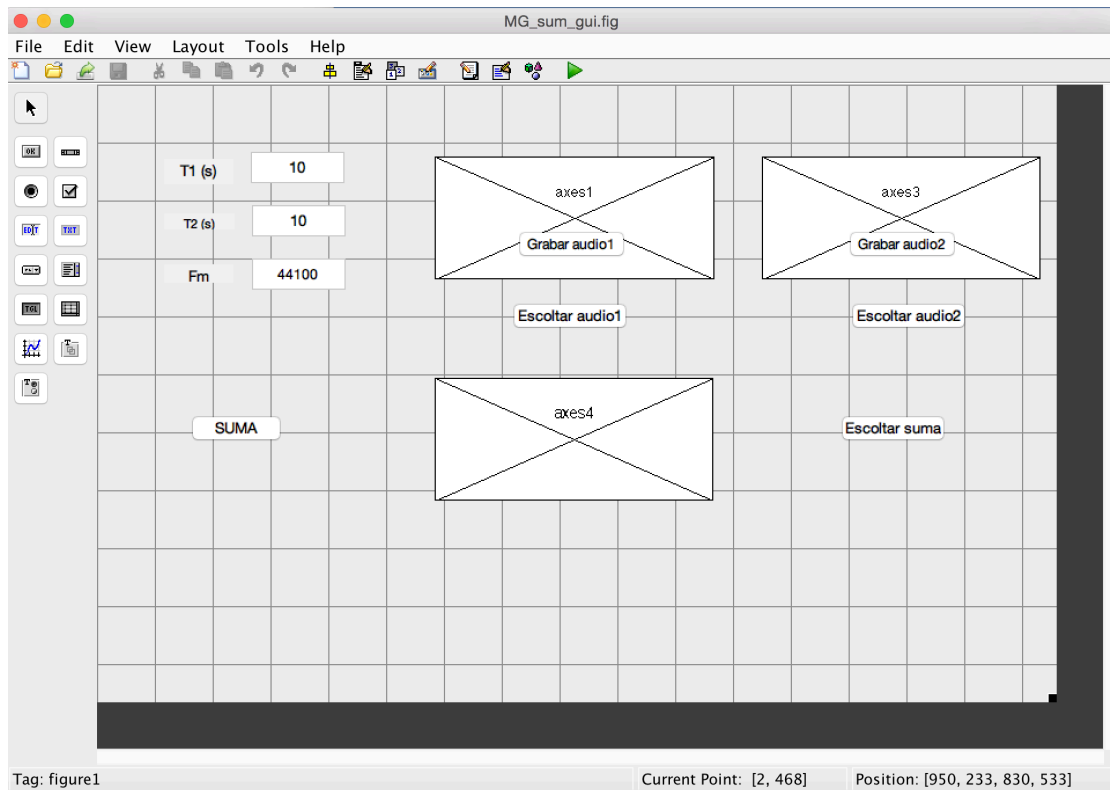


Figura 19: Disseny de suma de senyals

- T1/T2/Fm: Es tracta de tres Static text i tres Edit text, que el que permetran és poder introduir els paràmetres dels dos àudios que gravarem, corresponen als tag 'edit1', 'edit2', 'edit3'.
- Gravar audio1/audio2: Són dos botons, els quals permeten gravar els dos àudios amb el micròfon del PC, corresponen als tag 'pushbutton1', 'pushbutton2'.
- Gràfica 1/Gràfica 2: Representen els dos senyals d'àudio, serien el 'axes1' i el 'axes3'.

- Escoltar àudio1/àudio2: Són els botons encarregats de fer que s'escoltin els àudios, corresponen a 'pushbutton3' i 'pushbutton4'.
- Suma: És el botó encarregat d'operar amb els dos senyals sumant-los, es tracta del tag 'pushbutton6'.
- Gràfica 3: És 'axes3' i s'encarrega de representar la suma dels dos senyals.
- Escoltar suma: Amb aquest botó és possible escoltar la suma dels dos senyals, correspon al 'pushbutton7'.

A part de crear la figura, es crea una funció anomenada 'MG\_audio\_senyal.m' la qual, serà l'encarregada de poder gravar amb el micròfon cadascun dels àudios que sumarem més tard.

Els paràmetres d'entrada són el temps i la freqüència i retorna el senyal d'àudio enregistrat guardat en la variable x.

El procediment de la funció és llegir el que diem pel micròfon, amb la funció 'audiorecorder', i després amb el 'getaudiodata' la guarda en una variable.

També s'ha de crear el scrip principal, en el qual introduïrem les comandes, aquest farà possible el funcionament de l'aplicació, anomenat 'MG\_sum\_senyal\_gui.m'.

Per començar el que primer es fa es cridar a la figura perquè a l'inicialitzar el programa s'obri, tal com es mostra en el codi següent.

```
id=MG_sum_gui;
hid=guidata(id);
```

*Figura 20: Inicialitzar figura*

Seguidament, s'accedeix en una condició, si no existeix el primer senyal gravat 'x', es posen tots els elements que formen la figura en estat invisible, amb la funció set, excepte el botó de gravar el primer àudio i els paràmetres d'entrada, que els llegirem amb la funció get, la qual permet guardar el número introduït en una variable. A continuació, es pot veure al codi:

```

6      %set(hid.Text3,'visible','off');
7 -    set(hid.pushbutton2,'visible','off');
8 -    set(hid.pushbutton3,'visible','off');
9 -    set(hid.pushbutton4,'visible','off');
10 -   set(hid.pushbutton6,'visible','off');
11 -   set(hid.pushbutton7,'visible','off');
12 -   set(hid.axes1,'visible','off');
13 -   set(hid.axes3,'visible','off');
14 -   set(hid.axes4,'visible','off');
15 -   set(hid.axes1.Children,'visible','off')
16 -   set(hid.axes3.Children,'visible','off')
17 -   set(hid.axes4.Children,'visible','off')
18
19 -   fm=get(hid.edit3,'string');
20 -   fm=str2num(fm);
21 -   T1=get(hid.edit1,'string');
22 -   T1=str2num(T1);
23 -   T2=get(hid.edit2,'string');
24 -   T2=str2num(T2);

```

Figura 21: Funció set i lectura de variables

També crearem un vector anomenat “opc”, format al principi per tres zeros, que és el que ens permetrà anar avançant per l’aplicació.

En aquest moment, es pot clicar al botó de ‘Gravar àudio 1’, en el qual hem modificat el callback perquè en pressionar-lo utilitzi la funció esmentada anteriorment ‘MG\_audio\_senyal.m’ i gravi el senyal amb el corresponent micròfon. Per tant, el primer zero d’opc” passaria a ser 1, que significa que el primer pas de l’aplicació ja s’ha assolit

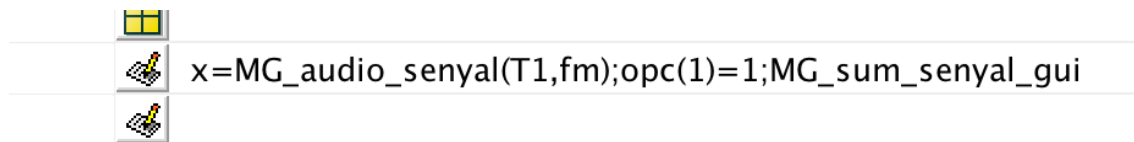


Figura 22: Funció MG\_audio\_senyal

Llavors ja existeix “x”, per tant amb la comanda set es visibilitzarà a la seva gràfica, el botó d’escoltar-lo, que conté la funció sound al callback, la qual permet que soni pels altaveus l’àudio gravat anteriorment, i el botó per a gravar el segon àudio. Es realitza el mateix procediment que anterior, s’ha de prémer el botó de ‘Gravar àudio 2’. En aquest moment, el segon element d’opc” passarà a ser 1 i un cop guardat a la variable y, apareix la seva gràfica, el seu botó per escoltar-lo i el botó de suma per continuar amb el procés de l’aplicació.



```

27 -         if opc(2)==1
28 -             set(hid.pushbutton2,'visible','off');
29 -             set(hid.pushbutton4,'visible','on');
30 -             set(hid.axes3,'visible','on');
31 -             set(id,'currentaxes',hid.axes3); % És n
32 -             plot(y), title('SENYAL 2')
33 -             set(hid.pushbutton6,'visible','on');

```

Figura 23: Gràfica d'un senyal

Finalment, quan es clica al botó suma, al callback cridem a una funció creada, anomenada 'MG\_suma.m', la qual té com a paràmetres d'entrada els dos senyals gravats, i com a sortida el senyal creat al sumar les anteriors. La seva funció és comparar les mides que tenen els dos senyals, amb la comanda 'size', per tal que la més petita s'ompli de zeros al final i passi a ser de la mateixa mida que la més llarga, igual que també mira si els dos senyals estan en estèreo o en mono, en el primer cas, es transformen, es fa la suma dels dos canals i es divideix entre dos per poder convertir-lo en un, tal com es veu al codi.

```

13
14 -     [n1,m1]=size(y1);
15 -     [n2,m2]=size(y2);
16 -     if m1==2, y1=(y1(:,1)+y1(:,2))/2; end
17 -     if m2==2, y2=(y2(:,1)+y2(:,2))/2; end
18 -     n=max(n1,n2)+1;
19 -     y1(n)=0; % posen la posicio n a 0 llavo
20 -     y2(n)=0;
21
22
23 -     x=y1+y2; %suma
24
25

```

Figura 24: Funció suma

En el cas que es vulgui canviar qualsevol dels senyals, només cal modificar els paràmetres de temps i de freqüència, si es canvia T1, es tornarà a gravar automàticament el primer àudio, es conservarà en segon àudio, i es tornarà a executar tot el programa. En el cas del T2 passarà al contrari que anteriorment, es torna a gravar l'àudio 2, i si es canvia la freqüència, llavors es gravarà primer l'àudio 1 i després l'àudio 2.

### 2.3.2. Filtres digitals

Per començar a desenvolupar l'aplicació de filtratge de senyals, es va començar creant els directoris i fitxers que es necessitarien. En el cas d'aquesta aplicació es necessiten 3 figures, i els seus corresponents 'scrips', que corresponen a cadascuna de les pantalles que utilitzarem, i diverses funcions les quals cadascuna tindrà el seu propòsit en aquest projecte.

- MENÚ

Primerament el que es realitza és el disseny de com es veurà la pantalla del Menú, amb tots els elements necessaris per a realitzar correctament la seva funció, en aquest cas crea un fitxer anomenat 'MG\_MENU.fig', que correspon a la figura que modificarem.

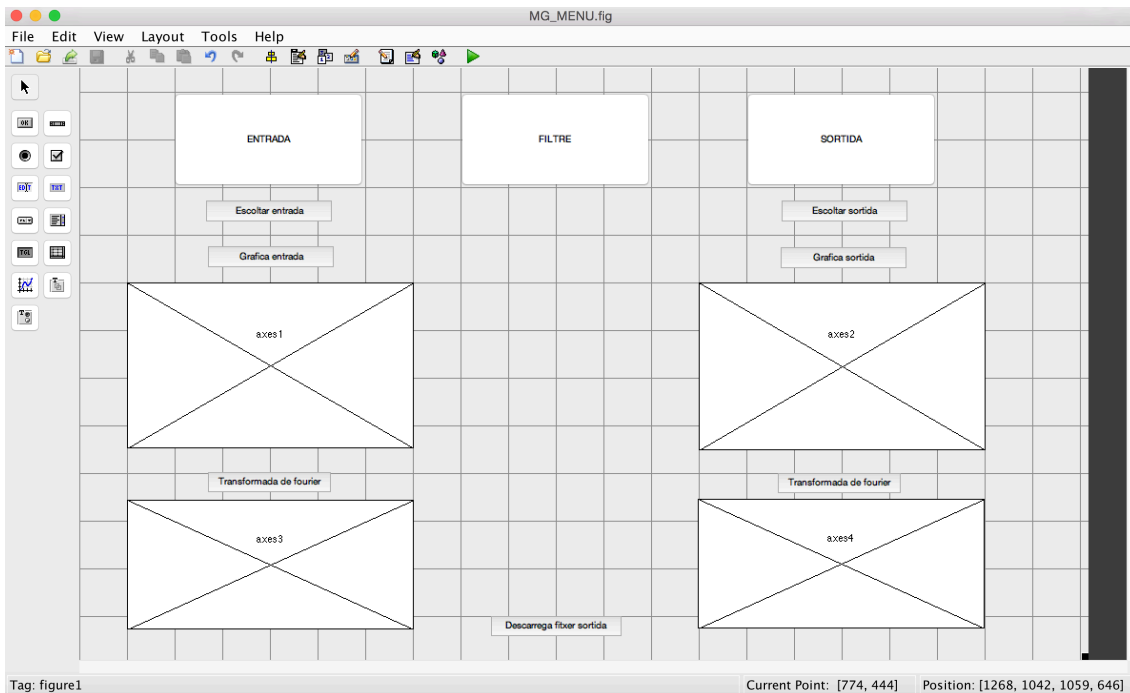


Figura 25: Disseny del Menú

Tal com es pot veure a la figura està compost per diversos tipus d'elements:

- ENTRADA: És un botó que es correspon amb el tag 'pushbutton4', el qual tindrà la funció de canviar a la pantalla d'entrada.
- FILTRE: És un botó, amb el tag 'pushbutton3', el qual tindrà la mateixa funció que ENTRADA, però per entrar a la pantalla de disseny del filtre.

- SORTIDA: Està assignat a 'pushbutton2' i es tracta del títol de la sortida, la qual correspon a l'entrada filtrada.
- Escoltar entrada/sortida: Es tracta de dos botons assignats als tags 'pushbutton4' i 'pushbutton8', el qual ens permetrà escoltar l'àudio d'entrada i de sortida.
- Gràfica entrada/sortida: És un botó assignat al tag 'axes1' i 'axes2', que representarà el senyal d'entrada i de sortida.
- Transformades de Fourier: Corresponen a 'axes3' i 'axes4', el botó activa l'acció de marcar una part de la gràfica del senyal d'entrada, i llavors es representa la transformada d'aquest tros a la segona gràfica.
- Descarregar fitxer de sortida: Correspon al tag 'pushbutton11', es tracta d'un botó el qual permet guardar l'àudio de sortida, és a dir el senyal d'entrada filtrat. Es posarà en estat visible quan ja s'hagi creat el senyal de sortida.

Per a assignar a cada element la seva funció, el que es fa és modificar el Callback de cadascun d'ells i crear un 'scrip' anomenat 'MG\_MENU\_AUDIO.m', on podem especificar el que aquesta part de l'aplicació anomenant-los amb el tag.

L'objectiu d'aquesta pantalla és bàsicament poder crear un senyal d'entrada, poder dissenyar un filtre, i amb aquests dos elements crear un senyal filtrat de sortida. També es permet comparar el senyal d'entrada amb el senyal de sortida, ja que es poden escoltar ambdós i veure la seva representació això com la seva transformada de Fourier.

En el 'scrip' el primer que es fa és cridar a la figura amb el seu identificador i obrir-la.

En el cas que no existeixi la variable d'entrada ni la funció de transferència del filtre, posem tots els elements invisibles, amb el 'set', menys el botó ENTRADA i el botó FILTRES, ja que de moment, la resta no s'utilitzaran perquè no tenim senyal ni es pot filtrar.

Qualsevol de les dues opcions, tant dissenyar el filtre com guardar el senyal d'entrada, es poden fer indiferentment en l'ordre que sigui necessari.

Per guardar el senyal d'entrada cal entrar al botó ENTRADA, el qual ens redirigirà a la pantalla d'entrada, en la qual es guardarà el senyal que es vol filtrar, es realitzaran els passos pertinents, que s'esmentaran més endavant, i per últim es torna al menú per poder seguir amb el procés de filtratge.

Aquesta redirecció a la pantalla correspon es realitza esmentant al callback del botó ENTRADA el programa relacionat amb la figura, tal com es veu al codi, i es torna al MENÚ.








ButtonDownFcn			
CData			
Callback		MG_ENTRADA_AUDIO	
CreateFcn			
DeleteFcn			

Figura 26: Callback per canviar de pantalla

Per tant en aquest moment, quan ja tenim senyal, el que passa és que apareixen visibles tots els elements relacionats, per tant es podrà escoltar l'àudio, en el cas que es pressioni el botó d'escoltar, ja que afegirem la funció 'sound' al seu callback, i es representarà el senyal, amb la funció 'plot'.

A continuació, per realitzar la transformada de Fourier, cal triar unes quantes mostres, això es podrà fer amb el punter, amb la funció 'ginput', la qual permet seleccionar dos punts del senyal i guardar-los en dues variables.

Aleshores es fa la transformada de Fourier del senyal que hi ha entre un punt i l'altre i es representa a la segona gràfica. Per activar aquesta funció cal clicar el botó de la figura de 'Transformada de Fourier'.

Per una altra banda, també s'ha de crear la funció de transferència, amb els paràmetres indicats, cal entrar a l'opció FILTRE, el qual es redirigirà a una altra pantalla, també esmentada posteriorment, llavors es tornarà a la pantalla MENÚ.

Finalment quan ja s'obtenen les dades necessàries per a poder filtrar el senyal d'entrada, es realitzarà la convolució entre el senyal i la funció de transferència.

Aquest fet generarà un senyal de sortida, que es podrà guardar, el qual correspondrà a l'entrada filtrada, per tant es posaran els elements restants apareixeran visibles, es podrà escoltar la sortida, veure la representació i tal com s'ha esmentat anteriorment, com a l'entrada, es podrà seleccionar amb el punter una part del senyal, per a fer la transformada de Fourier i la seva representació.

- ENTRADA

Una vegada el menú està dissenyat, el següent pas consisteix en dissenyar la pantalla ENTRADA, la qual apareixerà quan es pressioni el botó corresponent. Es crea la figura i se li anomena 'MG\_ENTRADA.fig'.

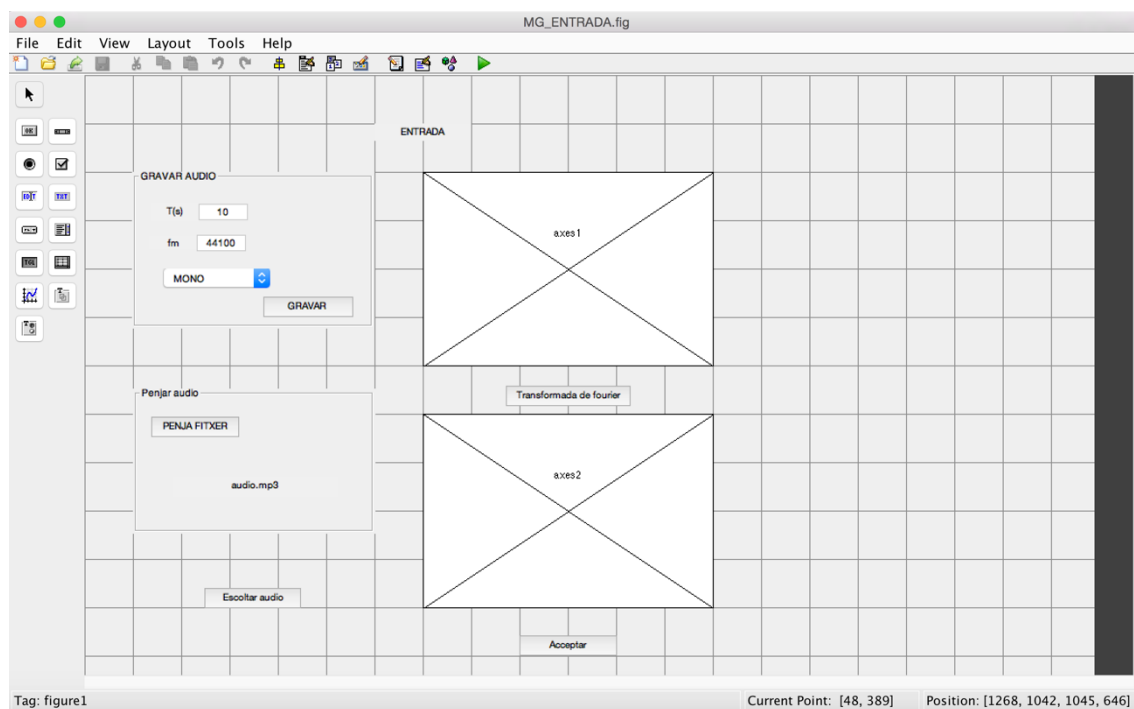


Figura 27: Disseny d'Entrada

Els elements utilitzats són els que es poden veure a la figura anterior, a l'hora d'adquirir l'àudio d'entrada, es pot fer de dues maneres, o gravant o penjant el fitxer:

- T(s)/Fm: Static text i Edit text, els quals serviran per saber el temps que dura l'àudio i la seva freqüència de mostratge, per poder gravar-lo, corresponen al tag 'edit1' i 'edit2'.
- MONO: Pop-up menú, el qual correspon a 'popupmenu1' i que permet triar en quants canals es vol gravar.

- GRAVAR: Botó que permet gravar l'àudio, amb el tag 'pushbutton2'.
- PENJA FITXER: Botó, el qual permet triar un fitxer guardat al PC, corresponen al tag 'pushbutton3'
- ESCOLTAR: Botó, amb el tag 'pushbutton4', el qual permet escoltar el senyal d'àudio que s'ha gravat o s'ha penjat.
- Gràfica: Corresponent al 'axes1', permet representar el senyal d'entrada.
- Transformada de Fourier: Correspon al 'pushbutton6' i a 'axes2', el botó activa l'acció de marcar una part de la gràfica del senyal d'entrada, i llavors es representa la transformada d'aquest tros a la segona gràfica.
- Acceptar: Botó que permet tornar a la pantalla del MENÚ, una vegada ja està l'àudio guardat, correspon al tag 'pushbutton7'.

En el cas de l'ENTRADA també s'ha utilitzat la funció anomenada 'MG\_audio\_senyal.m', esmentada anteriorment, la qual és la que permet gravar l'àudio des del micròfon del PC, retorna el senyal d'entrada, sabent el temps i la freqüència. Aquesta funció farà el seu objectiu en el moment que es pressioni el botó de gravar.

Aquesta pantalla es pot obrir des del menú, tal com s'ha explicat abans, clicant al botó d'ENTRADA.

A l'inici del programa principal 'MG\_ENTRADA\_AUDIO.m', el que es fa és cridar a la figura, perquè s'obri. Seguidament entrariem en una condició en la qual si no existeix el senyal d'entrada, llegim els paràmetres, de temps i de freqüència, i el pop-up menú, tal com es mostra al codi, per saber quants canals tenim, per si es necessiten per gravar, i també es posen els elements d'escoltar i representar el senyal a invisibles, ja que no s'estan utilitzant.


```

- fm=get(hid2.edit2,'string');
- fm=str2num(fm);
- T=get(hid2.edit1,'string');
- T=str2num(T);
- tpg=num2str(get(hid2.popupmenu1,'value'));
-
- switch tpg
- case '1'
-     canal=1;
- case '2'
-     canal=2;
- end

```

Figura 28: Funcionament tpg

Tal com s'ha esmentat anteriorment, a part d'obtenir el senyal d'entrada amb la gravació, es pot obtenir penjant el fitxer. Per tant seria el següent pas dins de la condició que encara no existeix el senyal d'entrada. Es modifica el callback del botó, afegint la funció 'uigetfile', que el que fa és obrir l'explorador de l'ordinador, per buscar la carpeta on es troba l'àudio, marcar-lo i es guarda com a valor de sortida de la funció el nom de l'arxiu i el destí on es troba. Això serveix a l'hora d'obtenir la ruta d'on es troba el fitxer, per buscar-lo i guardar-lo en una variable, amb la funció 'audioread'.



```

clear x; opc=1; [archivo,ruta]=uigetfile('*.mp3','*.wav');MG_ENTRADA_AUDIO

```

Figura 29: Escollir arxiu guardat al ordinador

```

23 - if opc==1
24 -     [x, fm]=audioread([ruta,archivo]);
25 -     if min(size(x))==2
26 -         x=(x(:,1)+x(:,2))/2;
27 -     end
28 -     set(hid2.text3,'String',archivo);
29 - end

```

Figura 30: Llegir arxiu del ordinador

Una vegada ja existeix el senyal d'entrada, es posen tots els elements en visible i podem escoltar l'àudio, quan se selecciona el botó corresponent, i es veu la representació a la gràfica utilitzant 'plot'.

En aquest cas és el mateix que en la figura del Menú, es pot triar una part del senyal per poder fer la transformada de Fourier i representar-la a la segona gràfica, aquest pas es pot fer tantes vegades com es vulgui, una vegada ja està la transformada del tros desitjat representada.

Finalment només cal que es cliqui a acceptar per tornar a la pantalla del menú i seguir amb tot el procediment. En el cas que es vulgui tornar a gravar un senyal d'entrada o penjar un altre fitxer, només es necessitarà clicar a qualsevol dels dos botons, el de 'gravar' o el de 'penjar un arxiu', i llavors es refrescarà el senyal i es tornarà a compilar tot el programa des del principi.

- FILTRES

L'última pantalla que s'ha de crear és la de disseny de filtres, 'MG\_FILTRES\_GUI.fig', composta pels elements que es veuen a la figura següent.

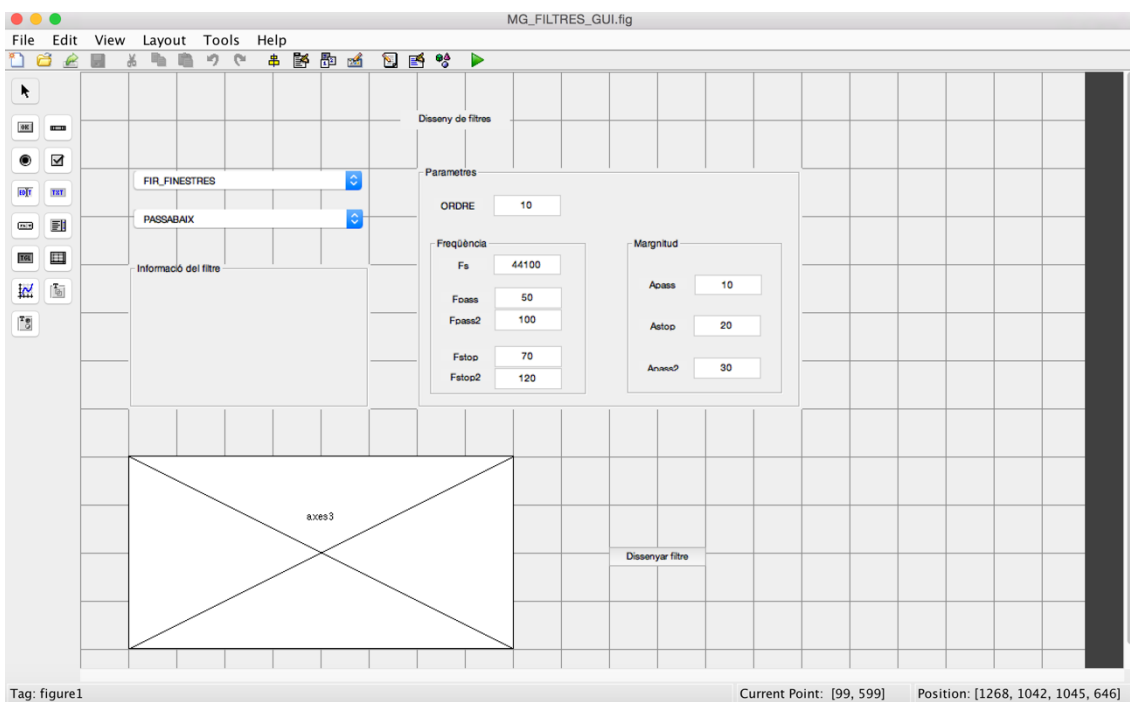


Figura 31: Disseny de Filtres

- FIR/IIR: És un pop-up menú que permet escollir entre si es vol que el filtre dissenyat sigui FIR o IIR, amb el tag 'popupmenu2'.



- PB/PA/PB/EB: Igual que l'anterior, però per triar si es vol que sigui un filtre passa-baix, passa-alt, passa-banda o elimina-banda; està assignat al tag 'popupmenu3'.
- Informació del filtre: És un quadre de text amb diversos Static text el que es mostraran les dades del filtre creat.
- Gràfica: Anomenat 'axes3', el qual ens representarà el filtre segons els paràmetres introduïts i la forma en què es vol veure.
- ORDRE/Fs/Fpass/Fstop/Fpass2/Fstop2/Apass/Astop/A2: Static text i Edit text, els quals es llegiran segons el tipus de filtre, permetran canviar les característiques del filtre i representar-los, corresponen als tags 'edit6', 'edit7', 'edit8', 'edit9', 'edit10', 'edit11', 'edit12', 'edit13'.
- Dissenyar filtre: Correspon el botó que farà possible guardar els valors del filtre dissenyat i tornar a la pantalla menú, correspon al tag 'pushbutton3'.

També es configura el Toolbar de la pantalla de filtres, tal com es veu a la següent figura, en el qual cada element nou correspon a una manera de representar el filtre que estem buscant, és a dir cada icona pertany a una de les opcions de representació esmentades més endavant.

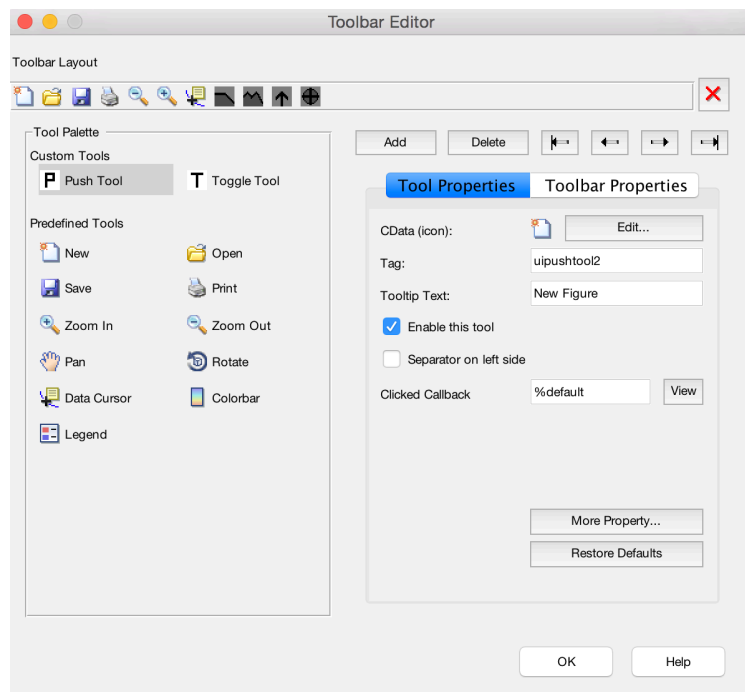


Figura 32: Disseny de toolbar de filtres

Es creen quatre icones noves, els tags corresponen a 'uipushtool1', 'uipushtool2', 'uipushtool3', 'uipushtool4', 'uipushtool5', la qual la primera representa la gràfica

en amplitud, la segona la gràfica en fase, la tercera la transformada de Fourier, la quarta seria la resposta impulsional del filtre i la cinquena els pols i zeros.

En aquest cas el programa principal és 'MG\_FILTRES.m' i el primer que fa, com els anteriors, és obrir la figura.

Seguint la teoria, depenent del filtre, es necessitarien diferents paràmetres, però en el cas d'ordre i de freqüència de mostratge es necessitarien pel disseny de tots, per això el primer que es fa es llegir aquests valors i guardar-los en variables.

Com per a cada tipus de filtre es necessiten uns paràmetres concrets, els elements que no s'utilitzen es posen en invisible. En el cas que els filtres siguin passa-baix o passa-alt, a part de l'ordre i la freqüència de mostratge, només es necessitarà la freqüència de pas, la d'atenuació i l'amplitud de pas i la d'atenuació, en canvi si es tracta d'un filtre passa-banda o elimina-banda, sí que es necessiten tots els paràmetres que es veuen a la figura 18.

Llavors el que es farà és llegir els dos primers pop-up i depenent de l'opció que hi hagi marcada, s'entrarà en cadascuna de les condicions corresponents. En el primer menú pop-up és pot triar si volem que el filtre sigui FIR òptim i IIR òptim, en canvi en el segon s'ha de triar si es vol filtre passa-baix, passa-alt, passa-banda o elimina-banda. Els FIR poden ser qualsevol dels altres quatre tipus, igual que els IIR. Llavors per a cadascun es tindrà una condició, és a dir només es tindran en compte els paràmetres necessaris, es guardaran a variables diferents i serviran de valors d'entrada de les funcions que ens retornaran per a cada tipus de filtre els coeficients,  $a$  i  $b$ , i els números de mostres, es pot veure tot el procediment en el següent codi.

```

17 - if tpg=='1'
18 -     %FIR_OPTIM
19 -     if tpg1=='1'
20 -         %PASSABAIX
21 -         set(hid3.edit8,'visible','off');
22 -         set(hid3.edit10,'visible','off');
23 -         set(hid3.edit13,'visible','off');
24 -         set(hid3.text8,'visible','off');
25 -         set(hid3.text10,'visible','off');
26 -         set(hid3.text14,'visible','off');
27 -         fpass=get(hid3.edit7,'string');
28 -         fpass=str2num(fpass);
29 -         fstop=get(hid3.edit9,'string');
30 -         fstop=str2num(fstop);
31 -         apass=get(hid3.edit11,'string');
32 -         apass=str2num(apass);
33 -         astop=get(hid3.edit12,'string');
34 -         astop=str2num(astop);
35 -         [b, M] = MG_FIR_optim_pb(fm, fpass, fstop, apass, astop);
36 -         [Hfir,F] = MG_trf(b,M);
37 -     end

```

Figura 33: Codi de variacions de variables segons el filtre

Les funcions creades per a cada tipus de filtre s'anomenen: 'MG\_FIR\_óptim\_pb.m', 'MG\_FIR\_óptim\_pa.m', 'MG\_FIR\_óptim\_pbn.m', 'MG\_FIR\_óptim\_pe.m', 'MG\_IIR\_óptim\_pb.m', 'MG\_IIR\_óptim\_pa.m', 'MG\_IIR\_óptim\_pbn.m' i 'MG\_IIR\_óptim\_pe.m'.

De cada tipus de filtre també és necessari calcular la funció de transferència, que ens servirà per poder fer les gràfiques corresponents, i per això es crea una funció 'MG\_trf.m', que amb els coeficients i el nombre de mostres ens la retorna i també la freqüència.

```

1  function [H,F] = MG_trf(b,M)
2  - if (length(M)>1) && ((M-round(M))~=0)
3  -     F=M(:); % Vector de freqüències en columna
4  - else
5  -     N=2*(M-1); % Nombre de punts en un període
6  -     F=(0:1/N:0.5)'; % Vector de freqüències, en columna (Nf per 1)
7  - end
8  - x=x(:); % Es treballa amb vector columna (Nx per 1)
9  - n=0:length(x)-1; % Índex temporal, en vector fila (1 per Nx)
10 - X=exp(-j.*2*pi.*F.*n).*b; % Vector columna amb la transformada
11 - end
12

```

Figura 34: Funció MG\_trf

Una vegada creat el filtre desitjat el que es fa és representar-lo a la gràfica, per comprovar que efectivament hem trobat les condicions que esperàvem, es podran triar diverses opcions possibles, esmentades anteriorment, llavors depenen del botó que es pressioni a la barra d'eines, la gràfica canviarà. Això és possible ja que, creen una variable, la qual cada vegada que es pressiona una de les possibles gràfiques canvia, per tant si l'opció es 1, es representarà el filtre en magnitud, si l'opció és 2 en fase, si l'opció és 3 es farà la transformada de Fourier i la seva representació, l'opció 4 es tracta de la resposta impulsional i la 5 dels pols i zeros.

Per últim, quan ja es troba el filtre ideal, es guarden els paràmetres necessaris per poder filtrar, clicant al botó de dissenyar filtre i es torna a la pantalla menú. En el cas que es vulguin canviar els paràmetres ja guardats anteriorment, es poden modificar tornant a entrar a la pantalla de disseny del filtre. Es modifiquen els paràmetres i es tornen a guardar.

### 3. RESULTATS

#### 3.1. PROVES

En aquest últim punt s'inclouran les explicacions i demostracions de les dues aplicacions creades.

### 3.1.1. SUMA DE SENYALS

Tal com es podrà veure a les figures següents, aquests són els passos que segueix l'aplicació. En la primera figura encara no es té cap dels dos àudios gravats, ens dona la opció de poder fer-lo amb el primer.

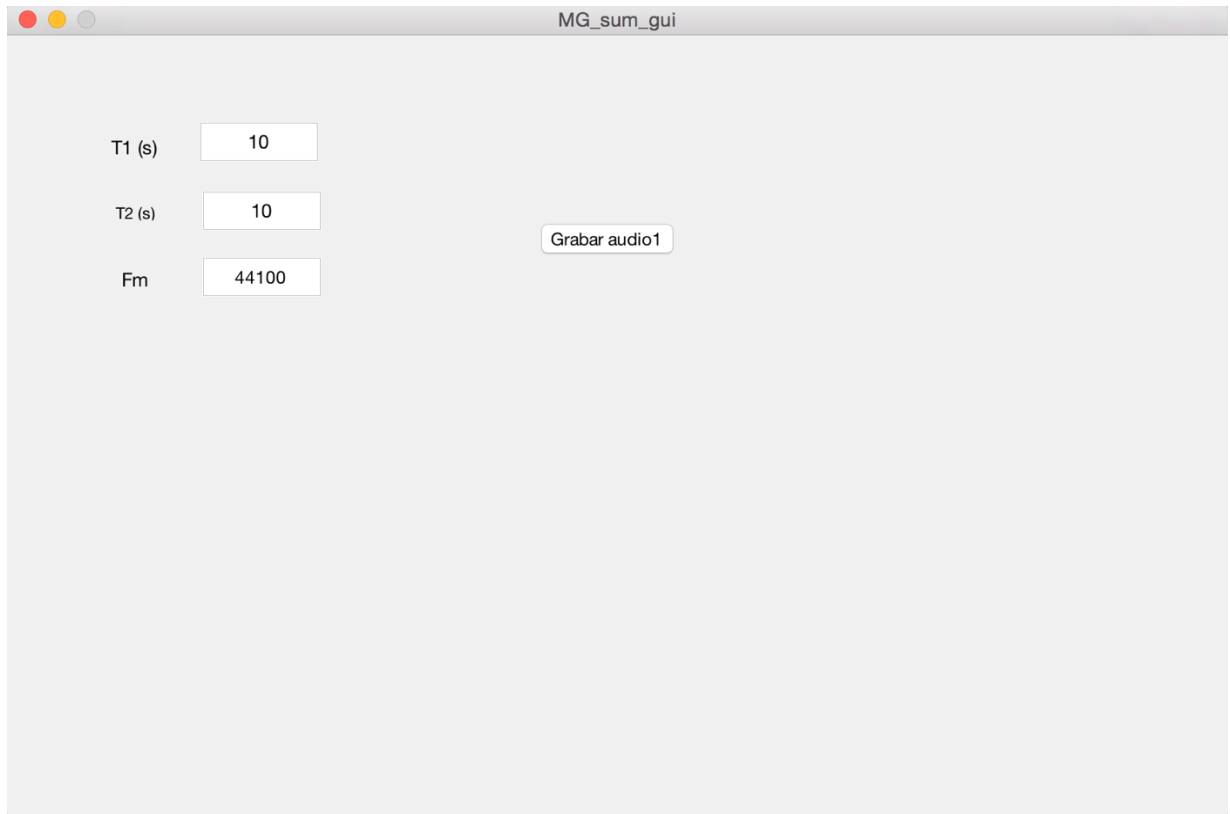


Figura 35: Gravar àudio 1

A la següent figura, ja s'ha gravat l'àudio 1, i es procedeix a gravar el segon àudio.

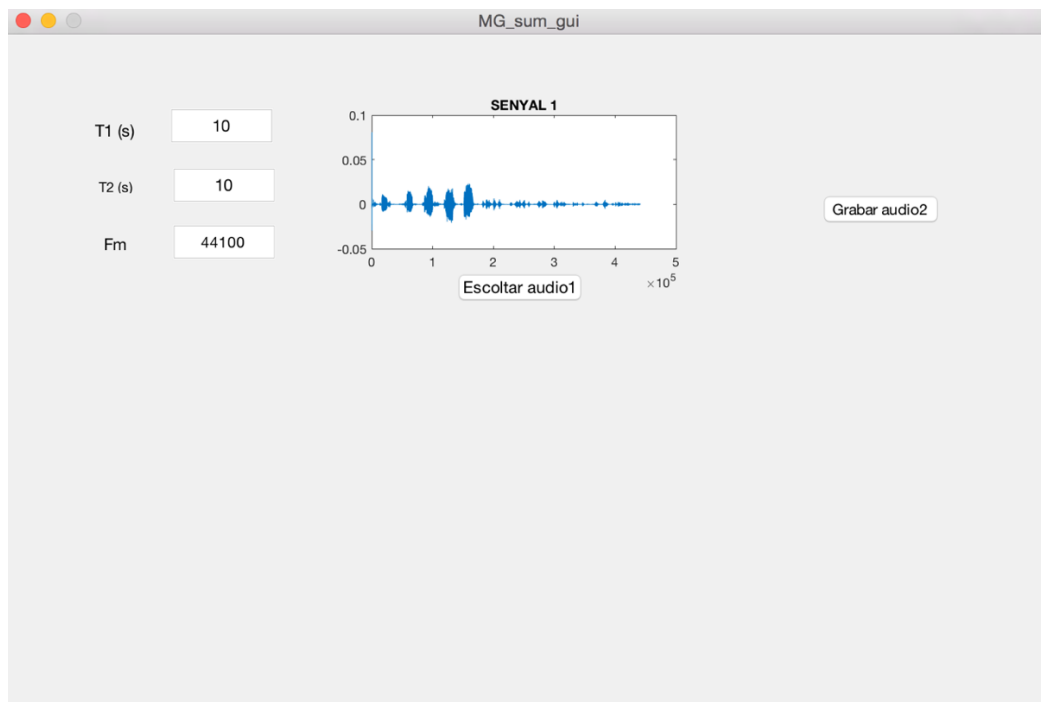


Figura 36: Escoltar i gràfica d'àudio 1 i gravar àudio 2

Una vegada ja tenim les dues senyals, apareix el botó de suma, que cridarà a la funció:

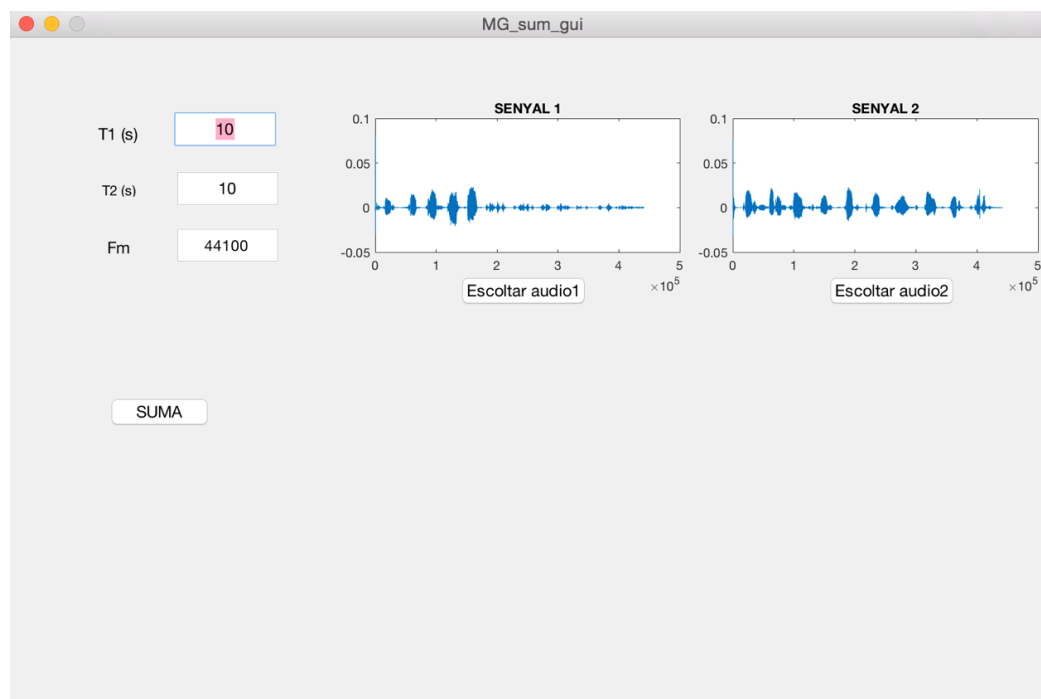


Figura 37: Escoltar i gràfica d'àudio 1 i àudio 2, botó suma

En aquesta última figura es pot veure el resultat.

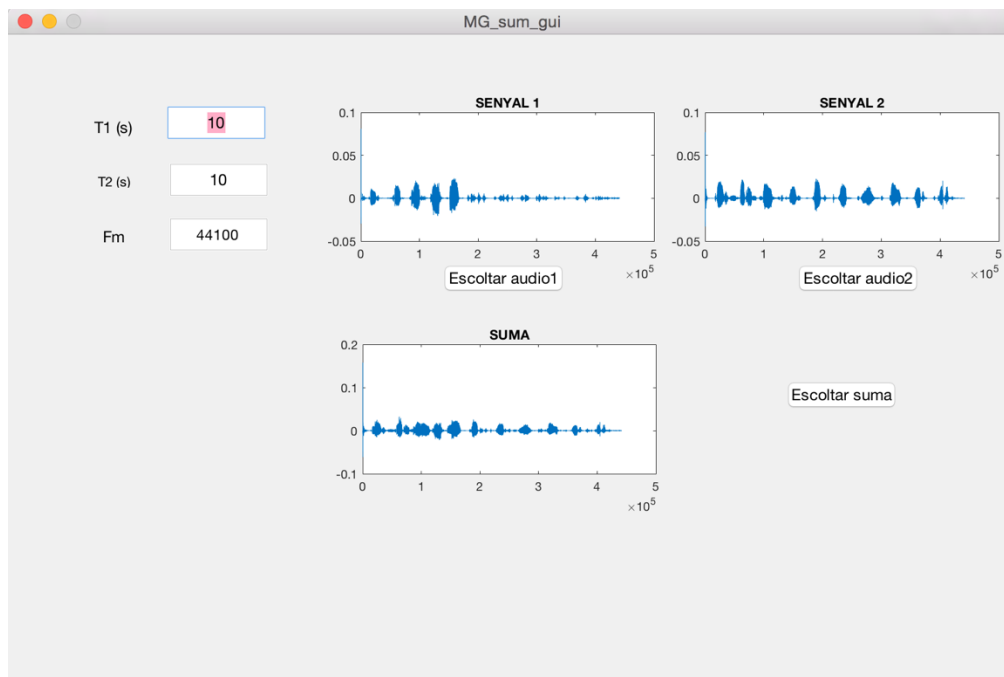


Figura 38: Escoltar i gràfica de la suma dels senyals

Tal com s'ha esmentat anteriorment, això és el que passaria si volem tornar a gravar, qualsevol dels àudios un altre cop, en aquest cas es torna a gravar l'àudio 1, es manté l'àudio 2 i s'esborra la senyal suma.

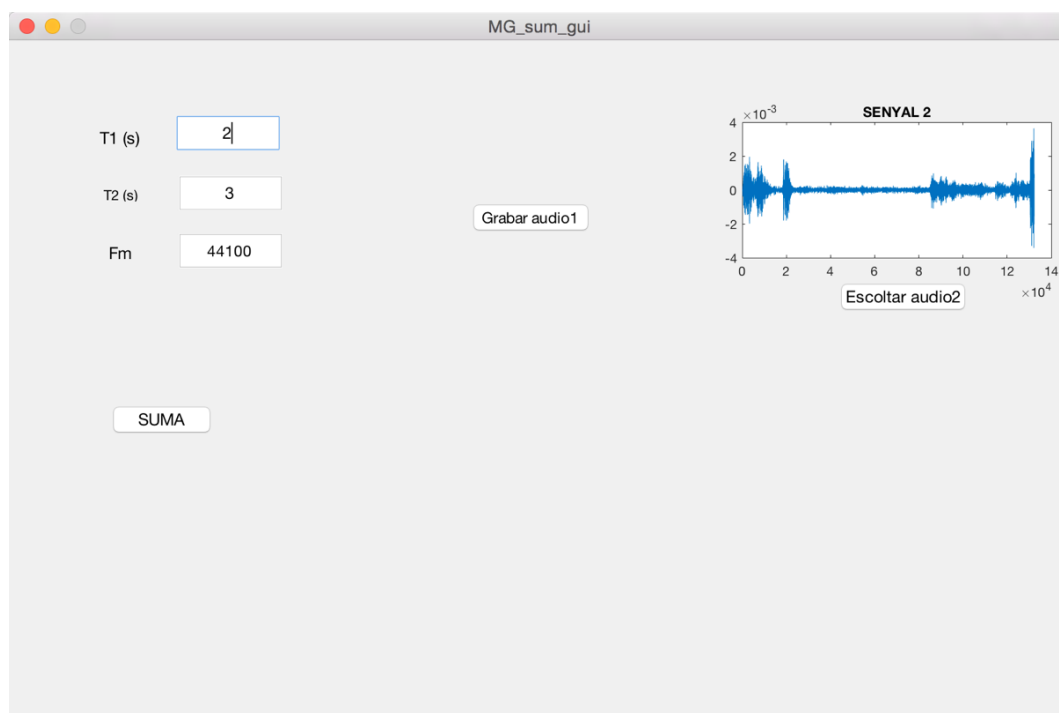
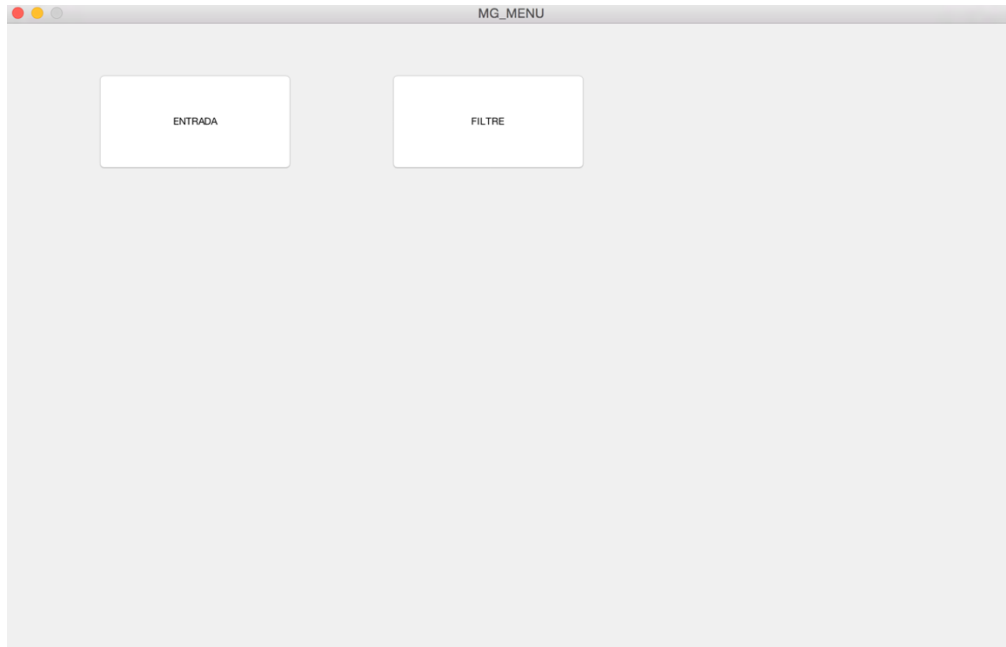


Figura 39: Repetir gravació d'àudio

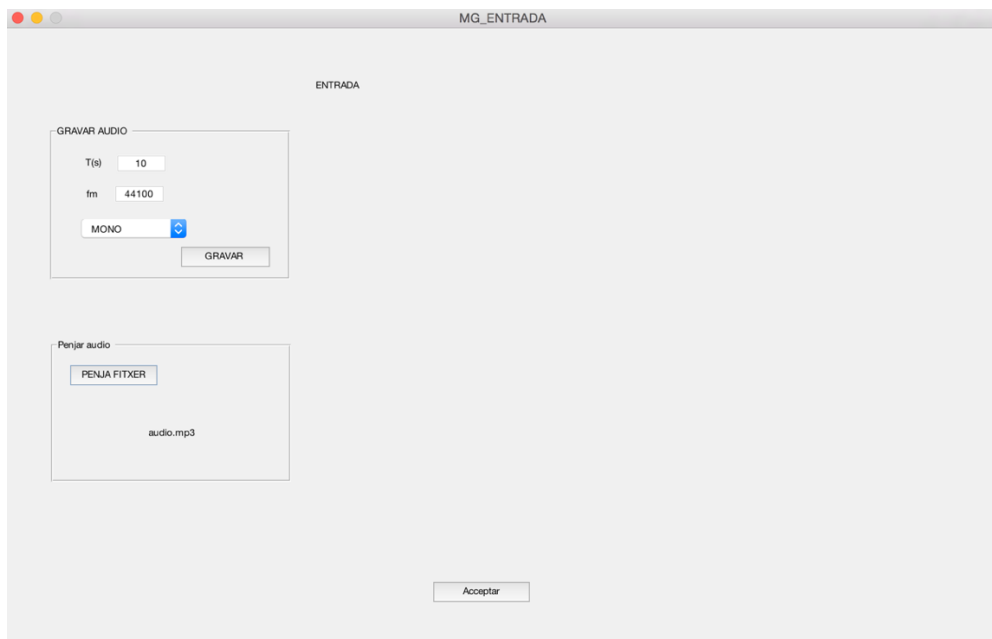
### 3.1.2. FILTRES

La primera pantalla de l'aplicació, quan encara no s'ha introduït el senyal d'entrada ni s'ha realitzat el disseny del filtre, és la següent:



*Figura 40: Menu aplicació disseny de filtres*

Quan es clica al botó d'ENTRADA es redirigeix a la pantalla corresponent que seria la següent:



*Figura 41: Pantalla d'entrada*



Llavors tal com s'ha esmentat anteriorment, hi ha la possibilitat de gravar l'àudio d'entrada o de penjar un fitxer, la qual es mostra en les dues figures que es mostren a continuació:

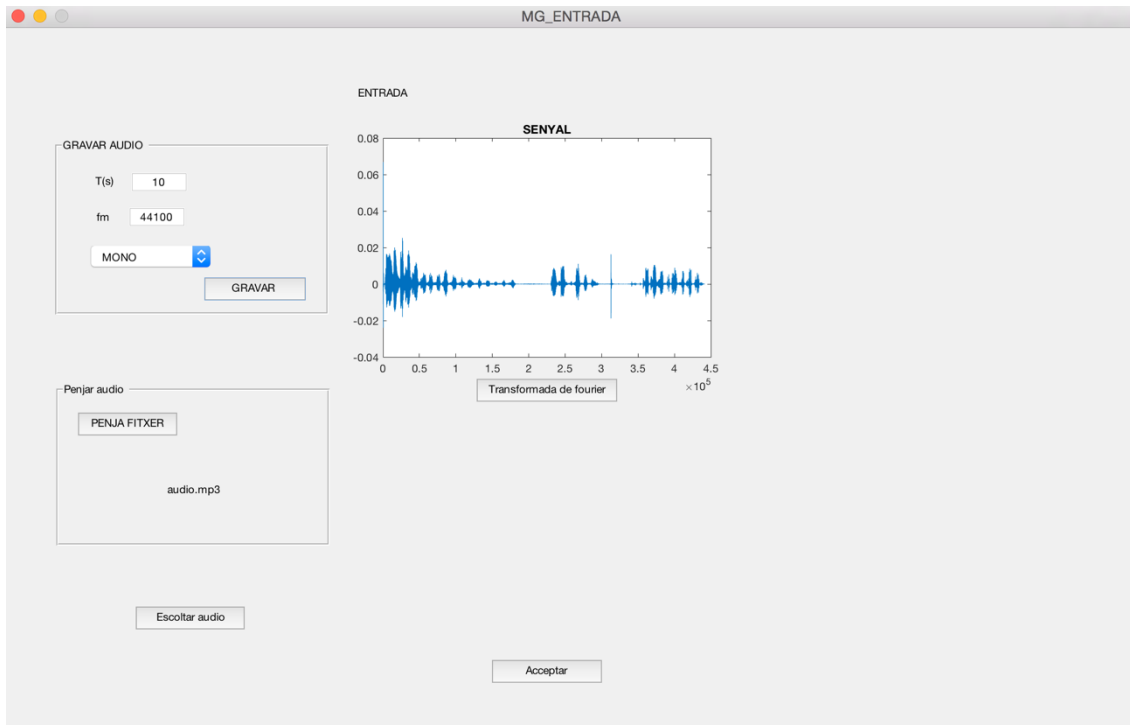


Figura 42: Gravar àudio d'entrada

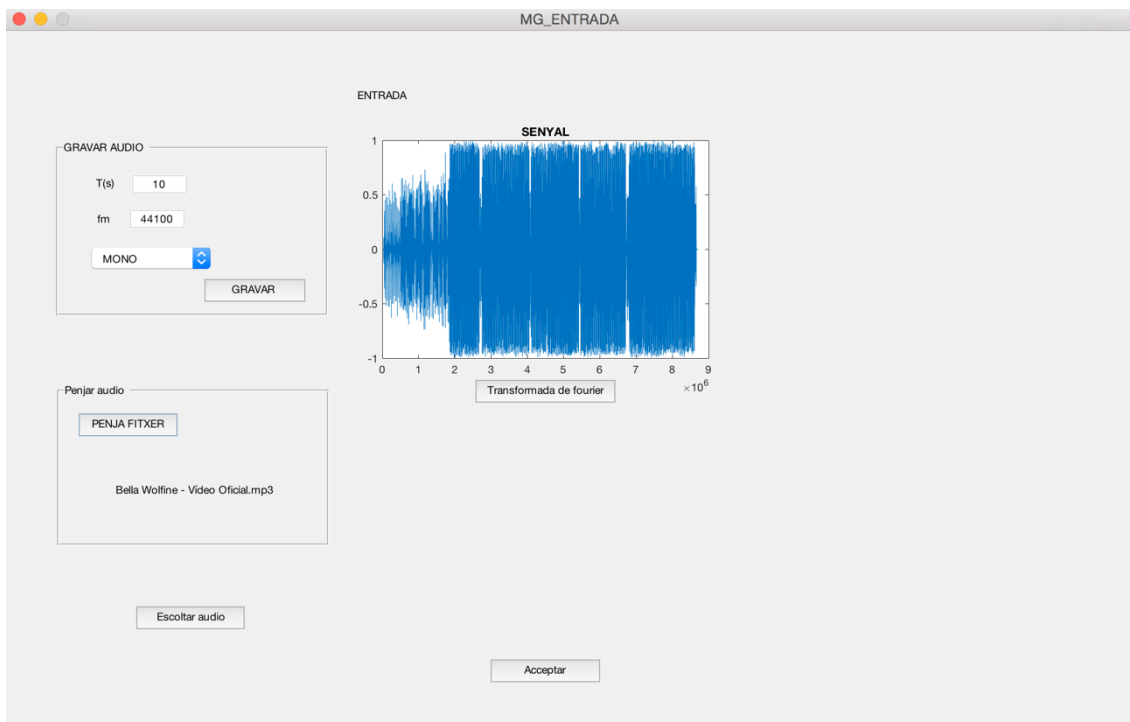


Figura 43: Penjar fitxer d'entrada

Es clica al botó transformada de Fourier i es passa a escollir dos punts de la gràfica, per després fer la seva presentació.

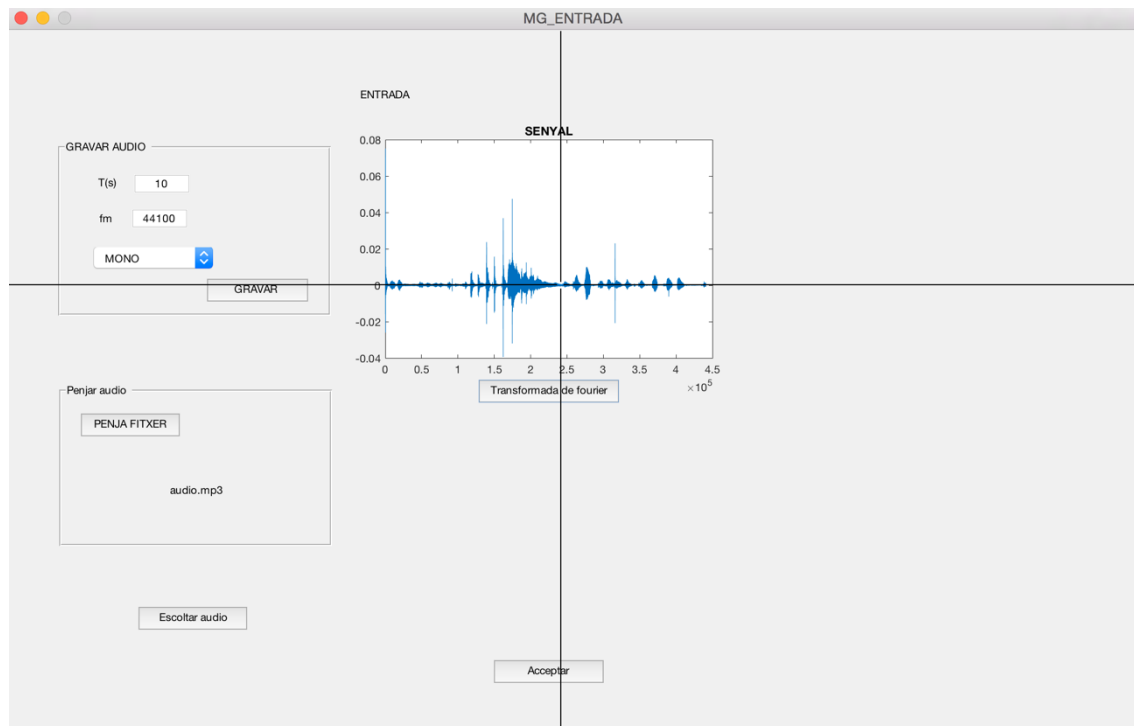


Figura 44: Elecció de punts per fer transformada de Fourier

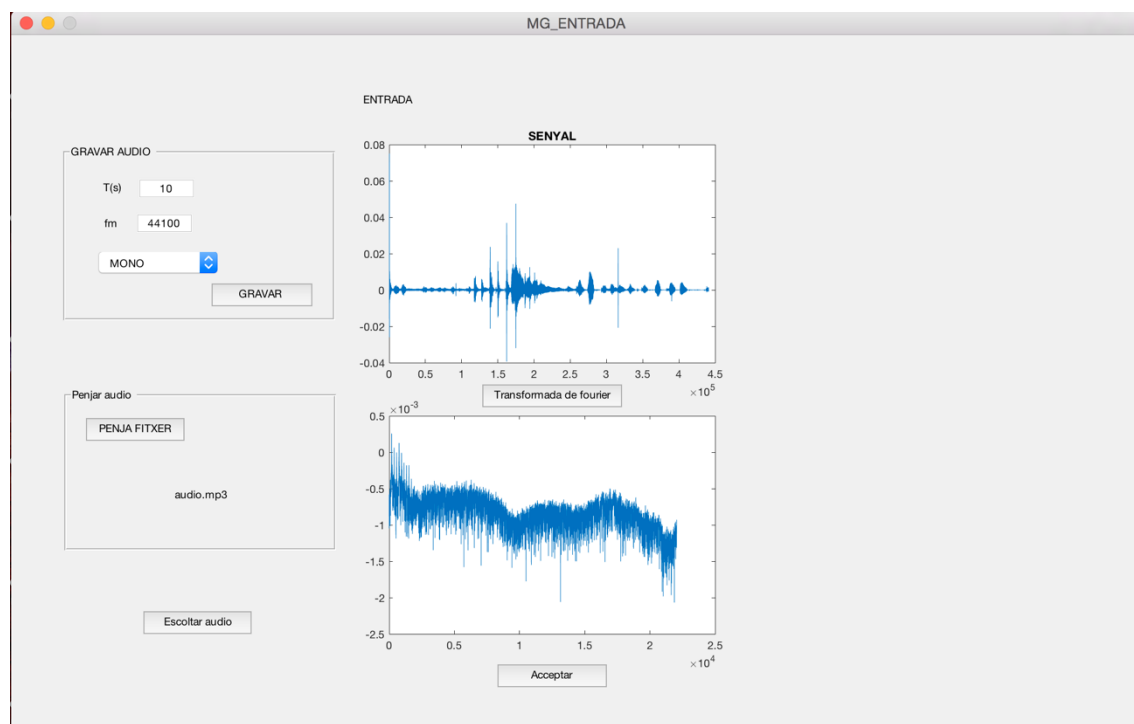


Figura 45: Transformada de Fourier de la senyal d'entrada

Una vegada ja estigui creat el senyal d'entrada, es clica a acceptar i es torna al menú.

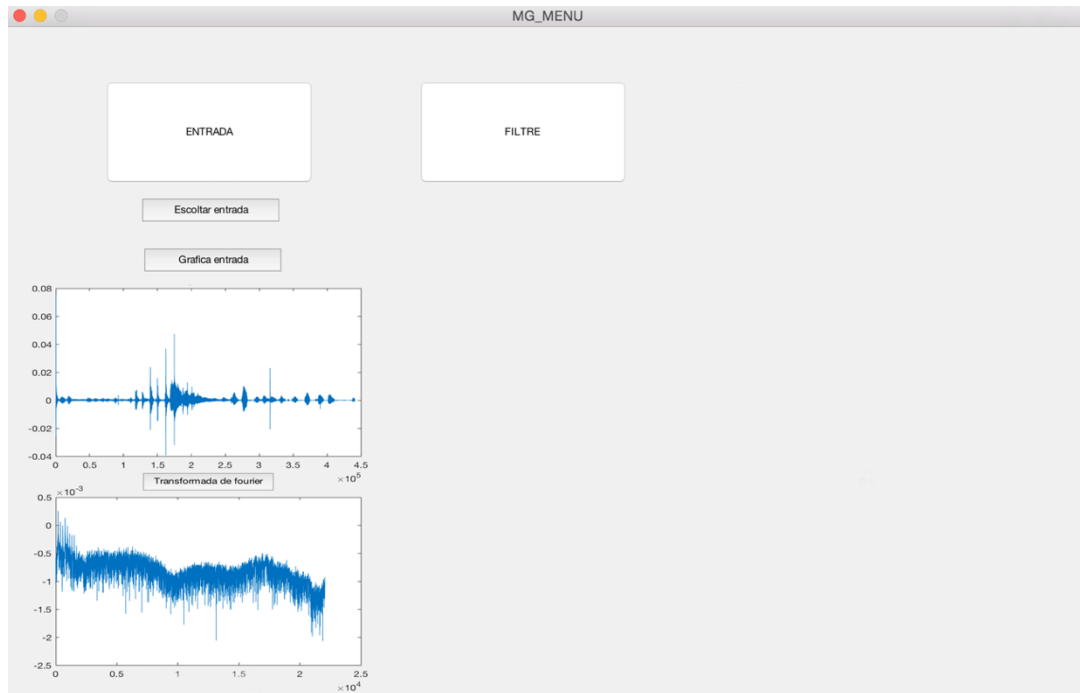


Figura 46: Menú amb la senyal d'entrada

Es clica a filtre i ens apareix en aquesta pantalla:

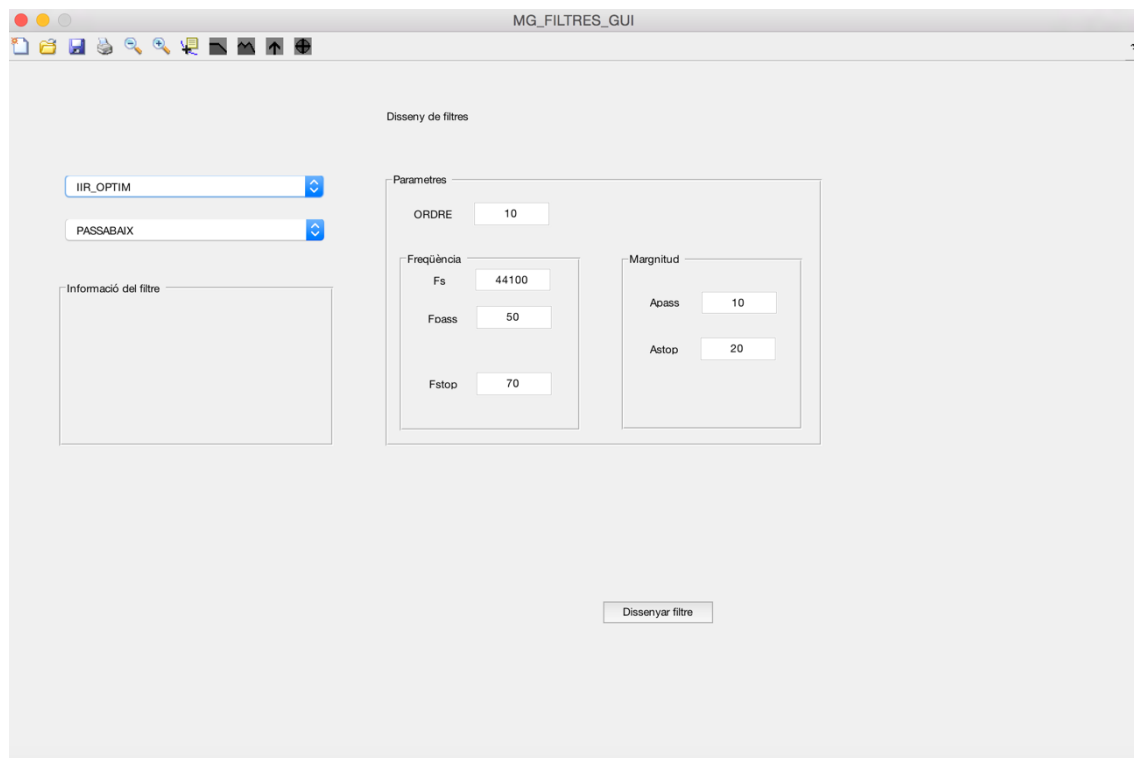


Figura 47: Disseny del filtre

En la qual el primer que es fa escollir entre les opcions dels diferents tipus de filtres, llavors varien els paràmetres d'entrada.

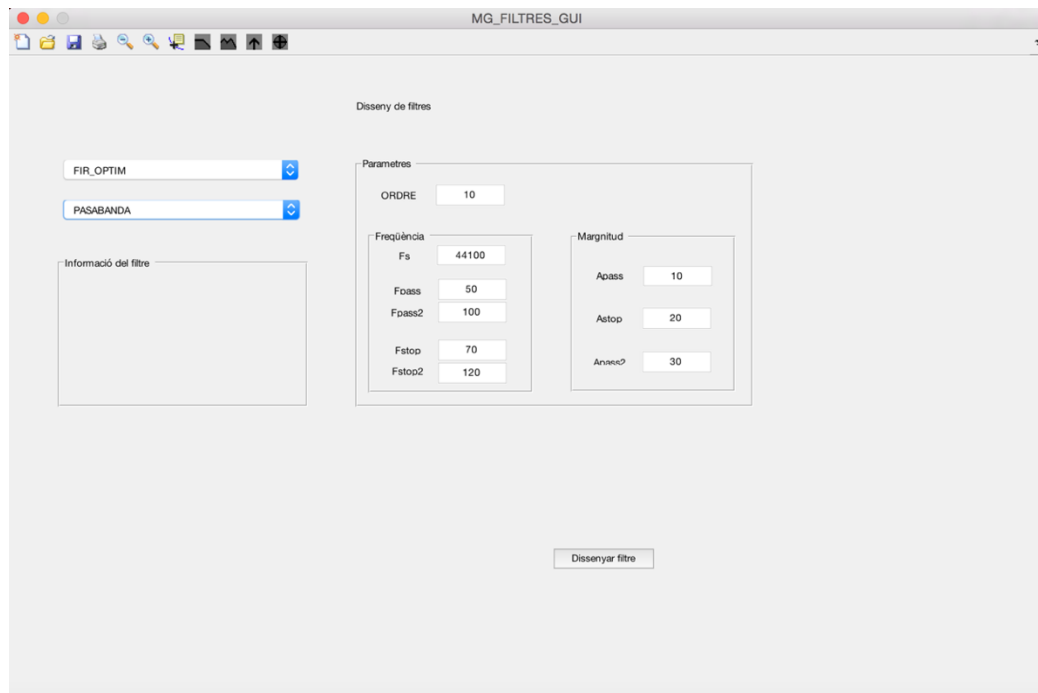


Figura 48: Canvi de paràmetres en disseny de filtres

Finalment quan ja sabem el tipus de filtres i els paràmetres, si se li clica a les icones de dalt, apareixerà la gràfica i la informació corresponent.

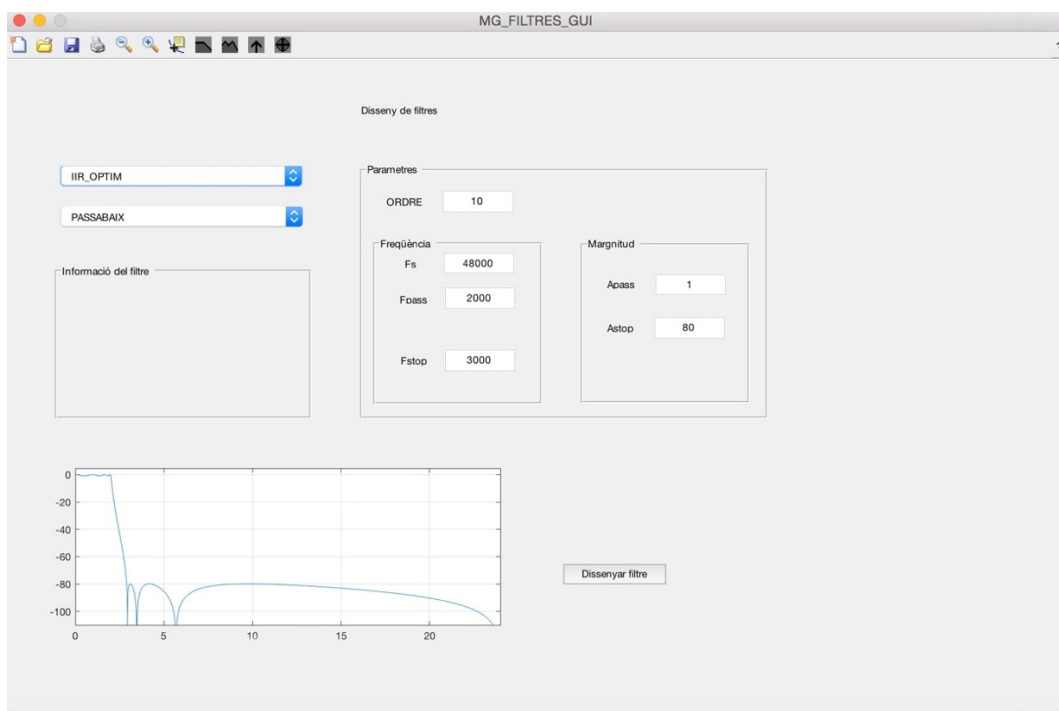


Figura 49: Gràfica en amplitud d'un filtre passa-baix IIR òptim

Llavors es torna al Menú, es clica al botó de sortida, i tal com s'ha esmentat abans, per mitjà de la convolució, es crea la gràfica del senyal filtrat i se li pot fer la transformada de Fourier d'un tros.

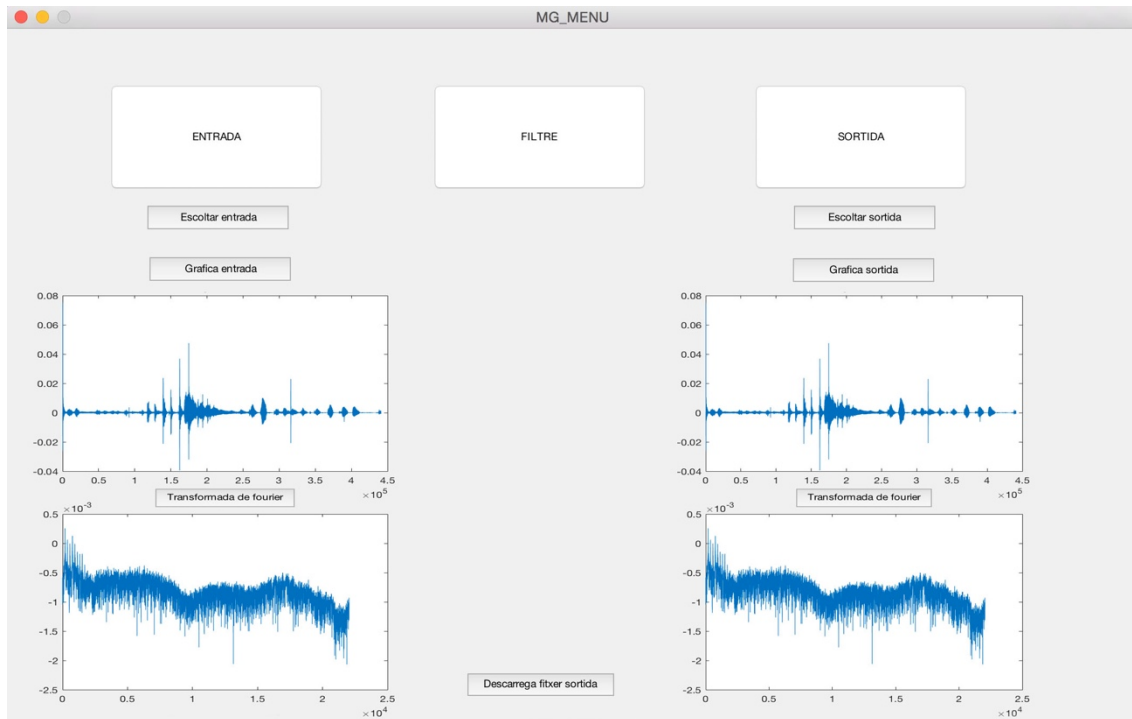


Figura 50: Menú amb la sortida

### 3.2. CONCLUSIONS

Amb aquest projecte es pretenia dissenyar i implementar una aplicació de filtres digitals, afegint l'opció de filtratge a l'aplicació de Filter Designer que ja existeix en el software de Matlab.

Després de tots els coneixements adquirits, tant de filtres com del disseny de l'entorn gràfic de Matlab, GUIDE, s'ha pogut realitzar amb èxit, ja que tots els propòsits esmentats a la introducció i al plantejament s'han desenvolupat.

El resultat ha estat una aplicació amb les funcionalitats de poder gravar o penjar un fitxer d'entrada, filtrar-lo, trobant els millors paràmetres i el tipus corresponent pel comportament desitjat, i així obtenir un senyal de sortida que compleixi els requisits buscats.

S'espera que aquest projecte sigui d'utilitat, tal com es diu a la introducció per a l'aprenentatge de la teoria dels filtres a la pràctica i per l'ús de gent amb coneixements, amb un resultat més precís que el que ja existeix, amb bons resultats de filtratge.

Finalment, podríem modificar o ampliar funcions de l'aplicació creada.

Un dels possibles canvis seria crear l'opció de poder gravar en estèreo, per dur a terme aquest canvi s'hauria d'afegir un altre element al disseny de la figura de ENTRADA, en la qual es grava l'àudio. El que s'hauria de fer seria afegir un pop-up menú en el qual es pogués escollir si el vol gravar en mono o en estèreo. Llavors dintre del 'scrip' principal de l'aplicació s'hauria d'incloure la lectura de l'opció escollida, i una condició en el qual si l'àudio és estèreo es fa la conversió a mono, procediment que ja es realitza en el cas de si es puja un fitxer d'àudio estèreo com a senyal d'entrada.

Un altre possible canvi, seria augmentar les opcions a filtres a escollir, és a dir com s'ha esmentat anteriorment, el que s'ha introduït dintre de l'aplicació són els filtres FIR i IIR òptims, però existeixen molts més tipus possibles per poder implementar. El necessari per incloure més varietat seria augmentar les opcions del primer menú desplegable de la pantalla de disseny de filtres, i crear per a cada filtre una funció específica perquè faci la seva funció.

#### 4. BIBLIOGRAFIA I WEBGRAFIA

1. Sisco Vallverdú. 'Apunts de filtres digitals'. Barcelona (UPC)
2. Roberts, M.J. 'Señales y sistemas: análisis mediante métodos de transformada y MATLAB'. México: McGraw Hill, 2005.
3. Oppenheim, A.V.; Willsky, A.S. 'Señales y sistemas'. 2a ed. México: Prentice-Hall Hispanoamericana, 1997.
4. Mathworks. 'Manuales de Matlab'. <https://es.mathworks.com>
5. Forester W. Isen. 'DSP for MATLAB and LabVIEW III'. 2008.

